

# DESENVOLUPAMENT D'UNA APP DE GOOGLE CHROME PER A LA DEFINICIÓ DE REQUISITS

Memòria del projecte

Autora: Núria Bertrán Hernández



Data Defensa: 28/01/2016

Directora: Carme Quer - ESSI

Especialitat: Enginyeria del Software

Titulació: Grau en Enginyeria Informàtica

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)– BarcelonaTech

## Agraïments

Vull expressar el meu agraïment a la meva directora del treball final de grau (TFG) Carme Quer i a l'estudiant de doctorat i cap del projecte del sistema PABRE Cristina Palomares, pels seus valuosos ajuts, suggeriments i direcció, i per la minuciosa correcció del meu treball.

Faig extensiu aquest agraïment als altres professors de la FIB que m'han proporcionat les habilitats necessàries per a dur a terme una tasca com aquesta.

Als meus pares i al meu germà pel seu ajut i encoratjament.

I, en general, a tots els amics que m'han donat suport al llarg d'aquests mesos de treball intensiu.

## Resum

Aquest projecte s'ha realitzat per permetre accedir al sistema PABRE de manera senzilla. PABRE és un sistema creat per ajudar als enginyers de requisits, aportant un catàleg de patrons de requisits que apareixen repetidament en projectes de software. En una base de dades hi ha emmagatzemat un catàleg de patrons que és accessible a través de serveis web. Els enginyers de requisits podran utilitzar-lo durant l'obtenció i especificació de requisits d'un projecte de software.

Aquest projecte permet a l'usuari accedir remotament a PABRE mitjançant una app de Chrome, la qual cosa permet una execució multiplataforma en els sistemes Windows, Mac, Chrome OS i Linux.

L'aplicació permet:

- Cercar patrons de requisits en el catàleg de PABRE mitjançant diverses vies.
- Afegir els requisits cercats a un editor de text inclòs a l'aplicació.
- Editar en l'editor de text l'especificació de requisits d'un projecte de software.
- Desar l'especificació de requisits com a projecte en el disc dur local.
- Recuperar especificacions de requisits locals desades anteriorment.
- Obrir especificacions de requisits d'exemple o plantilles d'especificacions que fan de punt de partida, ambdues incloses a l'aplicació.

## Resumen

Este proyecto se ha realizado para permitir acceder al sistema PADRE de manera sencilla. PABRE es un sistema creado para ayudar a los ingenieros de requisitos, aportando un catálogo de patrones de requisitos que aparecen repetidamente en proyectos de software. En una base de datos hay almacenado un catálogo de patrones que es accesible a través de servicios web. Los ingenieros de requisitos podrán utilizarlo durante la obtención y especificación de requisitos de un proyecto de software.

Este proyecto permite al usuario acceder remotamente a PABRE mediante una app de Chrome, lo que permite una ejecución multiplataforma en los sistemas Windows, Mac, Chrome OS y Linux.

La aplicación permite:

- Buscar patrones de requisitos en el catálogo de PABRE mediante diversas vías. Añadir los requisitos buscados en un editor de texto incluido en la aplicación.
- Editar en el editor de texto la especificación de requisitos de un proyecto de software.
- Guardar la especificación de requisitos como proyecto en el disco duro local.
- Recuperar especificaciones de requisitos locales guardadas anteriormente.
- Abrir especificaciones de requisitos de ejemplo o plantillas de especificaciones que hacen de punto de partida, ambas incluidas en la aplicación.

## Abstract

This project has been made to allow an easy access to the system PABRE. PABRE is a system created to help requirement engineers, providing a catalog of requirement patterns that appear repeatedly in software projects. A catalog of patterns is stored in a database, accessible via web services. Requirements engineers can use it for obtaining and specifying requirements of a software project. This project allows the user to remotely access PABRE using a Chrome app, which allows cross-platform systems running Windows, Mac, Linux and Chrome OS.

The application allows:

- Searching requirement patterns in the PABRE catalog in several ways.
- Adding the requirements sought in a text editor included in the application.
- Editing the requirements specification of a software project in the text editor.
- Saving the requirements specification as a project in the local hard drive.
- Recovering previously saved local requirements specifications.
- Opening requirements specifications or template specifications that make a starting point, both included in the application.

# Índex

<b>Agraïments .....</b>	<b>1</b>
<b>Resum .....</b>	<b>2</b>
<b>Resumen .....</b>	<b>3</b>
<b>Abstract .....</b>	<b>4</b>
<b>1 – Introducció .....</b>	<b>8</b>
1.1 – Contextualització .....	8
1.1.1 - Context .....	8
1.1.1.1 Parts interessades .....	11
1.1.2 - Estat de l'art .....	12
1.2 – Abast del projecte .....	13
1.2.1 - Formulació del problema .....	13
1.2.1.1 Objectius .....	13
1.2.2 – Abast .....	16
1.2.2.1 Possibles obstacles .....	16
1.2.3 - Metodologia i rigor .....	17
1.3 – Planificació temporal .....	17
1.3.1 - Descripció de les tasques .....	17
1.3.1.1 Fase Inicial .....	18
1.3.1.2 Desenvolupament .....	18
1.3.1.3 Fase final .....	19
1.3.1.4 Hores totals .....	19
1.3.1.5 Diagrama de Gantt .....	20
1.3.2 - Valoració d'alternatives i pla d'accions .....	20
1.4 – Gestió econòmica .....	21
1.4.1 - Identificació i estimació de costos .....	21
1.4.1.1 Costos directes .....	21
1.4.1.2 Costos indirectes .....	22
1.4.1.3 Consideració de riscos .....	24
1.4.1.4 Consideració per imprevistos .....	24
1.4.1.5 Resum de costos .....	25
1.4.2 - Control de gestió .....	25
1.5 – Sostenibilitat i compromís social .....	25
1.5.1 – Econòmica .....	26
1.5.2 - Social .....	26
1.5.3 - Ambiental .....	26
1.5.4 - Taula de sostenibilitat .....	27
1.6 – Estructura del document .....	27
<b>2 – Estudi del domini .....</b>	<b>28</b>
2.1 – Estructura dels SRPs .....	28
2.1.1 - Estructura d'un Patró de Requisits .....	28
2.1.2 - Patrons i esquemes .....	33
2.1.3 - Model conceptual de l'estructura .....	34
2.2 – Accés als patrons de requisits via PABRE-WS .....	35
2.3 – Model de casos d'ús .....	38

2.3.1 - Diagrama dels casos d'ús .....	38
2.3.2 - Cas d'ús: Iniciar app .....	38
2.3.3 - Cas d'ús: Editar requisits .....	39
2.3.4 - Cas d'ús: Guardar requisits .....	40
2.3.5 - Cas d'ús: Recuperar requisits .....	40
2.3.6 - Cas d'ús: Accedir als SRPs .....	41
2.3.7 - Cas d'ús: Insertar una part a l'editor .....	42
2.3.8 - Cas d'ús: Cercar els SRPs .....	42
2.3.9 - Cas d'ús: Seleccionar esquema .....	43
2.3.10 - Cas d'ús: Obrir templates .....	43
<b>3 – Estudi de la tecnologia .....</b>	<b>44</b>
3.1 – Estudi per desenvolupar Apps de Chrome .....	44
3.2 – Escollir software per l'aplicació .....	46
3.3 – Organització de l'entorn .....	49
<b>4 – Primera versió de PABRE-RW .....</b>	<b>51</b>
4.1 – Interfície de l'aplicació .....	51
4.2 – Escriure Requisits .....	55
4.3 – Guardar requisits .....	55
<b>5 – Segona versió de PABRE-RW .....</b>	<b>57</b>
5.1 – Editar els requisits .....	57
5.2 – Guardar els requisits .....	58
5.3 – Recuperar els requisits .....	59
5.4 – Interfície .....	59
<b>6 – Tercera versió de PABRE-RW .....</b>	<b>60</b>
6.1 – Afegir els SRPs .....	60
6.2 – Afegir els Forms dels SRPs .....	60
6.3 – Afegir les parts dels SRPS .....	60
6.4 – Usabilitat .....	60
<b>7 – Quarta versió de PABRE-RW .....</b>	<b>64</b>
7.1 – Cerca dels SRPs .....	64
<b>8 – Cinquena versió de PABRE-RW .....</b>	<b>65</b>
8.1 – Selecció d'una part d'un SRP .....	65
<b>9 – Sisena versió de PABRE-RW .....</b>	<b>67</b>
9.1 – Cercar un esquema .....	67
<b>10 – Setena versió de PABRE-RW .....</b>	<b>68</b>
10.1 – Selecció d'un projecte .....	68
10.2 – Interfície .....	70
<b>11 – Conclusions .....</b>	<b>71</b>
<b>12 – Glossari .....</b>	<b>74</b>
<b>13 – Referències .....</b>	<b>76</b>

<b>Annex: Índex de taules .....</b>	<b>78</b>
<b>Annex: Índex de figures .....</b>	<b>79</b>
<b>Annex: Anàlisi .....</b>	<b>80</b>
1 – Cas d'ús: Editar requisits .....	80
2 – Cas d'ús: Editar requisits .....	82
3 – Cas d'ús: Guardar requisits .....	89
4 – Cas d'ús: Recuperar requisits .....	90
5 – Cas d'ús: Accedir als SRPs .....	92
6 – Cas d'ús: Insertar una part a l'editor .....	94
7 – Cas d'ús: Cercar els SRPs .....	96
8 – Cas d'ús: Seleccionar esquema .....	98
9 – Cas d'ús: Obrir templates .....	99
<b>Annex: Manual d'instal·lació .....</b>	<b>101</b>
<b>Annex: User manual .....</b>	<b>103</b>



# 1 – Introducció

## 1.1 – Contextualització

### 1.1.1 - Context

El projecte s'emmarca dins el sistema PABRE [5] desenvolupat pel Grup d'Enginyeria del Software i dels Serveis (GESSI) de la UPC. Aquest sistema té com a objectiu ajudar als enginyers de requisits en la seva feina. Per fer-ho es basa en permetre gestionar un catàleg de patrons de requisits de software (SRPs) que l'enginyer de requisits podrà utilitzar durant l'el·licitació o obtenció i especificació de requisits d'un projecte de desenvolupament d'un sistema software.

La tasca més difícil de la construcció d'un sistema software, consisteix en determinar quin sistema s'ha de construir [6]. Aquesta tasca s'anomena Enginyeria de Requisits.

L'especificació de requisits es pot realitzar en diferents llenguatges o notacions. En el sistema PABRE s'ajuda a especificar requisits en llenguatge natural que descriuen el què ha de fer el sistema software (requisits funcionals) i com de bé ha de realitzar aquest les seves funcions (requisits no funcionals).

En el sistema PABRE els requisits que són recurrents en molts projectes de software es poden obtenir d'aplicar SRPs. La base de dades PABRE conté actualment un catàleg de SRPs que pot evolucionar amb el temps per incorporar més SRPs que s'identifiquin com a útils en els projectes.

Cada patró de requisits de software (SRP) conté uns elements anomenats formularis (*forms*), els quals al seu torn contenen *parts*, que poden ser fixes (*fixed*) o esteses (*extended*). Cada part conté un requisit genèric en llenguatge natural. Aquesta estructura permet agrupar uns patrons de requisits de manera ordenada.

L'utilització del sistema PABRE permet fer més eficient l'el·licitació de requisits, ja que l'enginyer de requisits no ha de perdre el temps en aquest tipus de requisits recurrents, i pot posar més atenció a altres requisits que donaran valor afegit al sistema software a desenvolupar. D'altra banda l'ús de SRPs farà que les especificacions de requisits siguin més completes i de més qualitat.

Actualment el sistema PABRE consta dels següents subsistemes [19]:

- PABRE-Man: Permet crear nous SRP i afegir-los a la base de dades. La base de dades de SRP és una base de dades MySQL situada en un servidor del departament d'Enginyeria de Serveis i Sistemes d'Informació (ESSI) de la UPC.

- PABRE-Proj: Permet escriure requisits basats en els SRPs. Aquests requisits es guarden a una base de dades de projectes que està situada al mateix servidor.
- PABRE-WS: Són serveis web REST que permeten a altres sistemes externs a PABRE accedir a la base de dades de SRP.

Dins del context del sistema descrit existeix un problema. Encara que PABRE-Proj ja és una aplicació per definir requisits a partir de patrons, aquesta aplicació és molt farragosa d'usar, té moltes funcionalitats, i requereix una corba d'aprenentatge molt gran. D'altra banda, obliga a tenir un coneixement molt gran de la base de dades de SRP. Finalment, guarda les especificacions de requisits en el servidor de la UPC (encara que es poden descarregar en una unitat local).

Aquest Treball Final de Grau (TFG) pretén donar solució a aquest problema, desenvolupant una aplicació:

- lleugera (només amb la funcionalitats mínimes)
- de fàcil ús
- d'aprenentatge fàcil
- que permeti guardar els requisits d'un projecte de manera local.

Com a requisits addicionals per a aquesta nova aplicació (que passarà a ser una més dins del sistema PABRE) es vol que estigui desenvolupada com una aplicació de Google Chrome. Aquest objectiu addicional, posat per la gent del grup GESSI, existeix per diferents motius:

- es vol explorar el potencial d'aquest nou tipus d'aplicacions
- es pretén esbrinar quines característiques específiques tenen pel que fa al desenvolupament i arquitectura.

La nova aplicació del sistema PABRE s'anomenarà PABRE-RW.

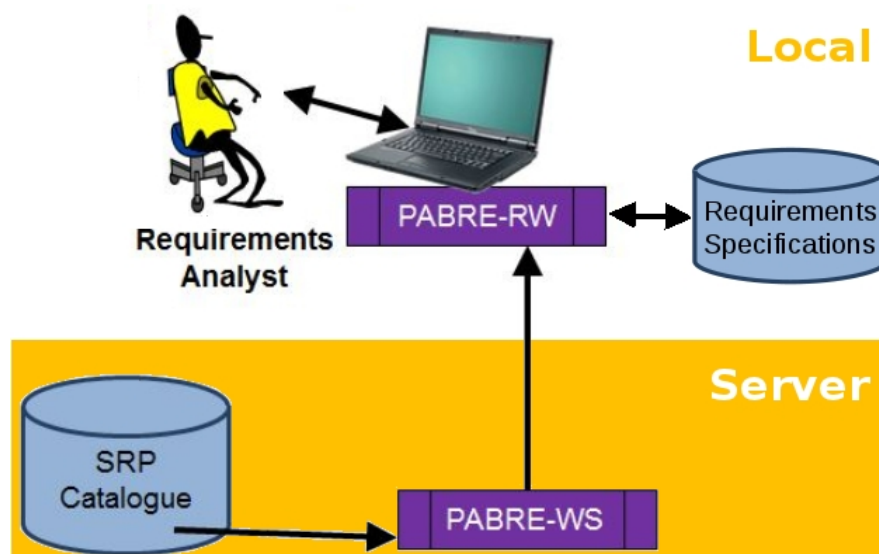


Figura 1: Disseny aplicació

D'aquesta manera el sistema PABRE, després d'afegir l'app PABRE-RW quedarà així:

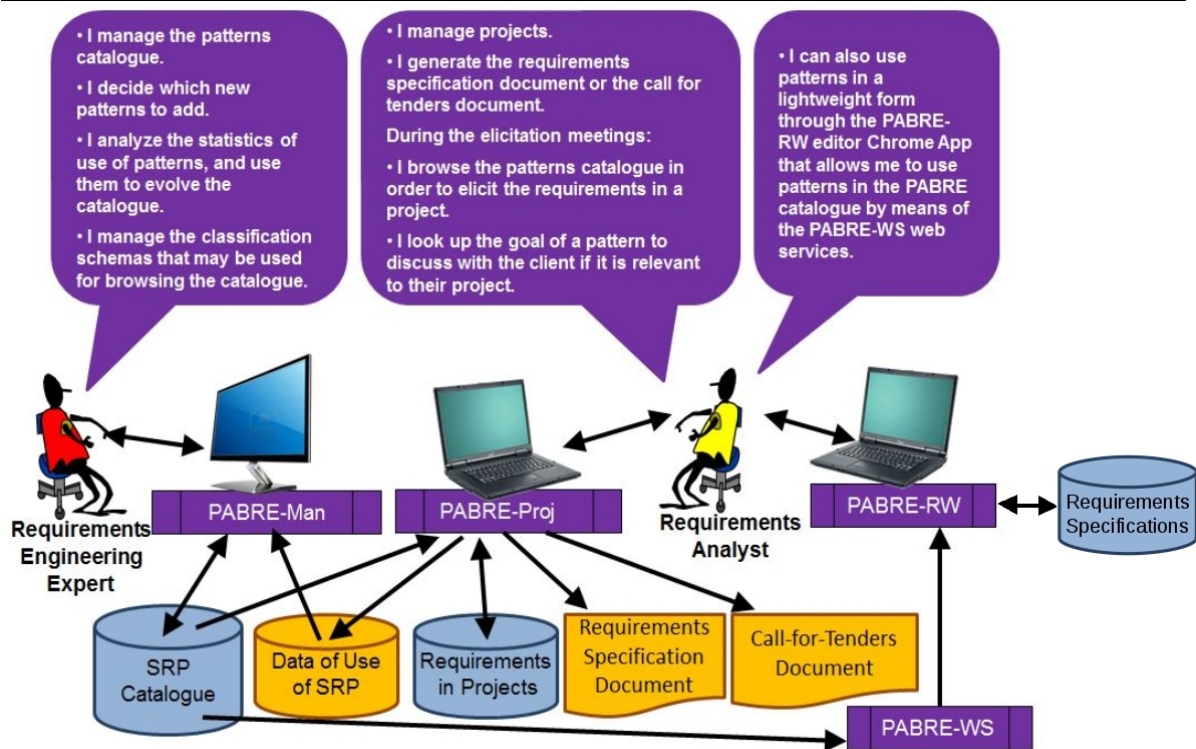


Figura 2: Disseny del sistema PABRE

Actualment hi ha dos tipus d'aplicacions de Chrome [7]

- Aplicacions que funcionen com llocs web normals (per exemple, Gmail o Google Calendar).
- Aplicacions que funcionen com programes d'escriptori i depenen de certes funcions de Chrome (com ara Google Keep).

Els avantatges d'aquestes aplicacions són els llistats a continuació [7]:

- Les aplicacions s'instal·len ràpidament, sense necessitat de reiniciar l'ordinador ni el navegador.
- Les aplicacions pròpies estan sempre disponibles quan es sincronitzen aplicacions en diversos ordinadors.
- Les aplicacions estan sempre al dia, ja que estan allotjades a Internet.
- Les aplicacions no bloquegen l'ordinador. Si hi ha problemes amb una aplicació, es pot tancar simplement la pestanya corresponent.
- És molt fàcil cercar i obrir l'aplicació amb el menú d'aplicacions.

En aquest TFG es pretén que PABRE-RW sigui una aplicació de Chrome que funcioni com un programa d'escriptori.

El resultat d'aquest TFG no es pretén que sigui un producte final, sinó un prototipus d'aplicació per definir requisits que haurà permès aprendre com es desenvolupen les aplicacions de Chrome, i també com s'utilitzen els serveis web REST.

#### 1.1.1.1 Parts interessades

A continuació es resumirà quines són les persones i institucions interessades en el TFG.

#### **Estudiant**

Com a estudiant que realitza el TFG, estic interessada en el projecte.

D'una banda, la realització d'aquest projecte suposa l'acabament dels meus estudis de grau.

D'altra banda, estic interessada en el projecte perquè amb la meva feina com a desenvolupadora de l'aplicació resultat del TFG, m'he encarregat de realitzar totes les tasques del projecte en els diferents rols que existeixen quan es desenvolupa software. Així doncs he après totes les tasques de l'enginyeria del software, que és la meva especialitat. I també he hagut d'aprendre noves tecnologies que no coneixia prèviament al projecte.

Finalment, també estic interessada en el projecte perquè m'ha permès millorar en com fer documentació de projectes informàtics, així com millorar en la redacció i presentació de treballs i memòries.

#### **Directora del projecte**

La directora del projecte és la professora Carme Quer. Ella és qui ha definit els requisits del TFG, com a membre del grup GESSI i com a una de les persones que ha participat en el desenvolupament del sistema PABRE. La cap del projecte del sistema PABRE Cristina Palomares també m'ha ajudat en la supervisió. Ella és també membre del grup GESSI, i està fent la tesis doctoral sobre SRP.

En ambdós casos els beneficis aportats durant el TFG permetran disposar d'una nova aplicació que passarà a formar part del sistema PABRE. D'altra banda, durant el TFG s'ha pogut fer el seguiment d'un projecte desenvolupat com una aplicació Chrome, aportant per tant al grup GESSI el coneixement necessari d'aquest nou tipus d'aplicacions, com es desenvolupen, etc.

#### **Enginyers de requisits**


Els enginyers de requisits són persones que es poden beneficiar de l'aplicació que es desenvoluparà en el TFG. L'existència de la nova aplicació PABRE-RW pot fer més fàcil l'ús de la base de dades de SRP, i per tant pot ajudar als enginyers de requisits a fer la seva feina de manera més eficient i amb més qualitat.

## Empreses que desenvolupen software

Les empreses que desenvolupen software es poden beneficiar de l'aplicació que es desenvoluparà en el TFG. L'existència de la nova aplicació PABRE-RW pot fer més fàcil l'ús de la base de dades de SRP, i per tant pot ajudar a que els enginyers de requisits que treballen en les empreses a fer la seva feina més eficient i amb més qualitat, reduint els costos i procurant un millor resultat dels projectes informàtics en que participen.

### 1.1.2 - Estat de l'art

D'aplicacions que permeten especificar requisits n'hi ha moltes. Les aplicacions de gestió de requisits existents més conegudes [3] es poden veure a la taula següent.

Aplicació	Aplicació de Chrome?	Lligada a la BD de PABRE?
SpiraTest [9]	X	X
ProcessStreet [10]	X	X
TraceCloud [11]	X	X
Gatherspace.com [12]	X	X
RequirementOne [13]	X	X
IBM Rational [14]	X	X
Innoslate [15]	X	X
Jama [16]	X	X
Enterprise Architect [17]	X	X
ReqView [18]		X

*Taula 1: Aplicacions de gestions de requisits*

La majoria no es limiten a gestionar requisits sinó que fan moltes altres tasques com ara gestió de workflows, SOPs, etc. A més, algunes, com Enterprise Architect, són eines no específiques de requisits, la qual cosa fa que treballar-hi els requisits sigui complicat. En general, això fa que l'ús d'aquestes eines tingui una corba d'aprenentatge alta, cosa que es vol evitar en l'aplicació PABRE-RW.

A la *taula 1* es poden veure dues columnes corresponents a les dues característiques que distingeixen l'aplicació PABRE-RW de les aplicacions existents:

- **Aplicació de Chrome?** Un requisit de PABRE-RW és que sigui una aplicació de Chrome. Com es pot veure a la taula, només ReqView ho és. Totes aquestes aplicacions necessiten una instal·lació independent, és a dir, no són una aplicació de Chrome i algunes ni tan sols van per web.
- **Lligada a la BD de PABRE?** Cal remarcar, però, que evidentment cap d'aquestes aplicacions no compleix un dels objectius bàsics del nostre projecte que és lligar a la base de dades PABRE. El fet de tenir els SRPs normalitzats en una base de dades centralitzada és un punt de vista novedós.

Pel que fa a ReqView [4] ja funciona com a aplicació de Chrome. Tot i que amb ReqView es poden gestionar els requisits, es presenta com un gestor de documents. A part de documents simples pot gestionar plantilles de documents que potser podrien fer el paper de SRPs, però a costa de molta més feina de la necessària degut a la complexitat de l'aplicació. L'aplicació ReqView es pot instal·lar des de Google Apps. Aquesta aplicació només deixa importar fitxers de Microsoft. Es pot exportar a formats PDF, HTML i MSWord. L'estructura del projecte de requisits resultant segueix una estructura d'arbre.

## **1.2 – Abast del projecte**

### **1.2.1 - Formulació del problema**

Tal com s'ha dit amb anterioritat, encara que el sistema PABRE disposa de l'aplicació PABRE-Proj i que aquesta aplicació ja permet definir requisits a partir de SRP, és molt farragosa d'usar, té moltes funcionalitats i requereix una corba d'aprenentatge molt gran. D'altra banda, obliga a tenir un coneixement molt gran de la base de dades de SRP. Finalment, des de les especificacions de requisits en el servidor de la UPC per defecte, la qual cosa complica el baixar-los a local.

Aquest TFG té com a objectiu solucionar aquest problema tot desenvolupant l'aplicació PABRE-RW. Aquesta nova aplicació va estar pensada per ser lleugera (només amb la funcionalitat justa), de fàcil ús, de fàcil aprenentatge, i que permetés guardar els requisits d'un projecte de manera local.

Com a requisits addicionals per aquesta nova aplicació (que passa a ser una més dins del sistema PABRE), ha estat desenvolupada com una aplicació de Chrome per poder explorar el potencial d'aquest nou tipus d'aplicacions i veure quines característiques específiques tenen pel que fa a desenvolupament i arquitectura.

#### **1.2.1.1 Objectius**

Els objectius a seguir en el projecte són:

##### **Objectiu 1**

Comprovar que es pot accedir als SRPs de la base de dades de PABRE mitjançant els serveis web de PABRE (PABRE-WS). Entendre l'estructura d'un SRP i l'estructura dels documents que retornen els serveis web. Entendre els elements d'un SRP i com s'usen els SRP en la definició de requisits.

La nostra aplicació accedeix a la BD de SRP de PABRE a través de l'API de serveis web REST definida pel grup de recerca GESSI a la web de PABRE [5].



## Objectiu 2

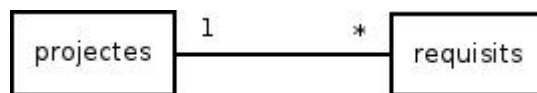
Estudiar la documentació existent sobre com desenvolupar aplicacions de Chrome. Escollir el software més adequat per al desenvolupament de l'aplicació. Analitzar la millor manera d'estructurar l'aplicació que es vol desenvolupar.

S'han consultat algunes pàgines web que serveixen com a guia per a la documentació sobre aplicacions de Chrome i sobre llenguatges i plataformes per desenvolupar aquestes aplicacions.

## Objectiu 3

Desenvolupar una primera versió de PABRE-RW que permeti escriure requisits, guardar aquests requisits en una base de dades local.

Aquesta base de dades ha de ser la representació del següent diagrama UML:



*Figura 3: Base de dades dels requisits d'un projecte*

S'ha necessitat estudiar el millor sistema per a emmagatzemar els requisits en l'ordinador des d'on s'usa PABRE-RW. Durant l'estudi de la tecnologia realitzat durant l'objectiu 2, s'ha donat preferència a conèixer les alternatives existents en aplicacions Chrome per emmagatzemar dades (ex: Chrome Storage API, aquesta API està definida a la pàgina web oficial de desenvolupadors de Google Chrome [2]).

## Objectiu 4

Desenvolupar una segona versió de PABRE-RW que permeti recuperar els requisits d'una base de dades local, editar-los i guardar-los altre cop modificats.

La interfície ha de permetre seleccionar un projecte concret, poder llistar i modificar els requisits que hi hagi en el projecte, així com crear de nous o eliminar alguns dels existents. A més, un cop acabades les modificacions pertinents, s'ha de poder guardar el projecte.

També cal poder crear i eliminar projectes. De fet això correspondrà a guardar un nou fitxer. En endavant es considerarà fitxer i projecte com a sinònims dins l'aplicació.

## Objectiu 5

Crear una tercera versió de PABRE-RW on s'afegeixi una nova àrea en la que es mostrin els SRPs i el contingut de cada un d'ells.

Durant l'execució de l'objectiu 4 s'ha desenvolupat una aplicació amb una única finestra on hi ha els requisits d'un projecte recuperats d'una base de dades local. Aquest objectiu consisteix en afegir una nova àrea a la interfície, on es mostraran tots els SRPs que hi ha a la base de dades PABRE. Per cada SRP cal que es llistin tots els seus continguts. A la interfície desenvolupada durant l'assoliment de l'objectiu 4 hi ha tota la informació per editar requisits en una columna. En aquesta nova versió es pretén afegir una nova columna a la seva dreta per a poder cercar els SRPs.

### **Objectiu 6**

Crear una quarta versió de PABRE-RW on s'afegeixin facilitats de cerca/ordenació dels SRPs i del contingut de cada un d'ells.

Es vol facilitar la feina a l'enginyer de requisits a l'hora de trobar el SRP que vol aplicar en un projecte. Això es fa oferint funcionalitats de cerca. La cerca d'un SRP es realitza tenint en compte les paraules que hi ha al goal, keywords, i description dels *form* i de les *parts* d'un SRP.

### **Objectiu 7**

Crear una cinquena versió de PABRE-RW on es pugui seleccionar una part d'un SRP i usar-la en la definició de requisits.

Aquest objectiu permet que l'usuari seleccioni una part d'un form d'un patró (la part és una frase en llenguatge natural), i aquesta part s'afegeix al projecte com a requisit. Durant l'aplicació, s'ofereix informació a l'usuari per tal que sàpiga com aplicar el SRP.

A més a més, al llarg del projecte i un cop assolit els objectius plantejats inicialment, s'ha anat més enllà i s'han afegit dos objectius més (objectius 8 i 9).

### **Objectiu 8**

Crear una sisena versió de PABRE-RW on s'afegeixi la cerca d'un esquema (*Classifier Schema*), que és un sistema jeràrquic de classificació dels SRPs.

En aquest objectiu es té en compte com estan classificats els esquemes i quins SRPs hi ha en cada un dels directoris de l'esquema.

Aquest objectiu inicialment no hi era, es va afegir a finals de desembre.

### **Objectiu 9**

Crear una setena versió de PABRE-RW on es puguin seleccionar uns projectes ja predeterminats que poden servir d'exemple.



Per l'assoliment d'aquest objectiu s'han creat uns fitxers, és a dir, uns projectes prefabricats per que l'usuari pugui veure l'estructura que hi tenen.

Aquest objectiu inicialment no hi era, es va afegir a finals de desembre.

### 1.2.2 - Abast

L'abast del projecte queda reflectit en els objectius concrets del TFG que ahora estan relacionats amb les iteracions que s'han anant fent durant el desenvolupament de PABRE-RW i que són:

1. Comprovar que es pot accedir als SRPs de la base de dades de PABRE mitjançant els serveis web de PABRE (PABRE-WS). Entendre l'estructura d'un SRP i l'estructura dels documents que retornen els serveis web. Entendre els elements d'un SRP i com s'usen els SRP en la definició de requisits.
2. Estudiar la documentació existent sobre com desenvolupar aplicacions de Chrome. Escollir el software més adequat per al desenvolupament de l'aplicació. Analitzar la millor manera d'estructurar l'aplicació que es vol desenvolupar.
3. Desenvolupar una primera versió de PABRE-RW que permeti escriure requisits, guardar aquests requisits en una base de dades local.
4. Desenvolupar una segona versió de PABRE-RW que permeti recuperar els requisits d'una base de dades local, editar-los i guardar-los altre cop modificats.
5. Crear una tercera versió de PABRE-RW on s'afegeixi una nova àrea en la que es mostrin els SRPs i el contingut de cada un d'ells.
6. Crear una quarta versió de PABRE-RW on s'afegeixin facilitats de cerca/ordenació dels SRPs i del contingut de cada un d'ells.
7. Crear una cinquena versió de PABRE-RW on es pugui seleccionar una part d'un SRP i usar-la en la definició de requisits.
8. Crear una sisena versió de PABRE-RW on s'afegeixi la cerca d'un esquema (*Classifier Schema*), que és un sistema jeràrquic de classificació dels SRPs.
9. Crear una setena versió de PABRE-RW on es puguin seleccionar uns projectes ja predeterminats que poden servir d'exemple.

#### 1.2.2.1 Possibles obstacles

El principal obstacle és que s'ha treballat amb uns llenguatges i entorns que ha calgut aprendre, i pels que hi ha hagut dificultats a l'hora de l'aprenentatge o a l'hora d'usar-los per desenvolupar l'aplicació PABRE-RW.

D'altra banda per començar el projecte també es requeria dedicar un temps d'investigació per saber quines eines eren les més adequades per treballar. A més s'havien de tenir en compte altres eines en cas de que les que s'han utilitzat fossin massa complicades.

Un projecte desenvolupat amb una tecnologia que d'entrada no es coneixia podria haver estat un obstacle per arribar a assolir tots els objectius enumerats. D'haver disposat de més temps, s'hauria eliminat o simplificat l'objectiu 6 que tracta de donar facilitats de cerca i ordenació dels SRP i les seves parts a la interfície.

### 1.2.3 - Metodologia i rigor

S'ha utilitzat la metodologia Scrum [8]. Per cada objectiu del projecte, s'ha descompost l'objectiu en les tasques a realitzar i s'ha assignat un temps necessari per realitzar aquestes tasques.

Un cop definides les tasques a fer, la Carme Quer (directora del projecte) i la Cristina Palomares (cap del projecte del sistema PABRE) han fet de Product Owner. Per cada tasca, l'han dividit en subtasques, han ordenat aquestes subtasques per ordre d'importància i preferència, tot creant el que es coneix com a Product Backlog, que és, en resum, el conjunt de requisits a alt nivell prioritzats que defineixen la feina a fer.

S'han fet Sprints, o cicles d'una setmana. És a dir, cada setmana s'ha fet una reunió entre la Carme Quer, la Cristina Palomares i jo mateixa on s'ha planificat el Sprint. Durant cada reunió:

- S'ha prioritzat la feina assignada al Sprint amb el Product Owner.
- S'ha considerat l'esforç necessari per realitzar cada element assignat al Sprint.
- En funció de l'esforç disponible (hores \* mida de l'equip \* durada del Sprint) es selecciona quina feina es fa per la següent setmana.

A l'inici de la reunió al final de setmana s'ha fet una revisió del Sprint per veure el progrés i així millorar el rendiment del treball de les properes setmanes. La revisió del sprint setmanal ha consistit en:

- Revisar la feina que s'ha pogut completar i la que no.
- Presentar la porció d'aplicació desenvolupada durant la setmana. Normalment amb una demostració.
- Parlar sobre els aspectes positius i els que ha calgut millorar per optimitzar el desenvolupament del TFG.

## 1.3 – Planificació temporal

### 1.3.1 - Descripció de les tasques

El projecte ha tingut una durada de 5 mesos. Ha començat a principis de setembre . La durada del projecte ha estat de 540 h, és a dir, les 33 h setmanals.

Per a fer la planificació hi ha hagut tres fases: la planificació inicial, el desenvolupament del projecte i la fase final. Per preveure possibles desviacions la fase de desenvolupament inicialment s'havia d'acabar a finals de desembre i d'aquesta manera tenir temps per fer modificacions, en cas que sigui necessari, i poder fer l'última fase.

#### 1.3.1.1 Fase Inicial

La fase inicial es fa en el curs del GEP i al llarg dels mesos de setembre i octubre. En aquesta fase s'inclouen les tasques següents:

- Definició de l'abast i contextualització
- Planificació temporal
- Gestió econòmica i sostenibilitat
- Presentació preliminar
- Document de la fase inicial
- Presentació de la fase inicial

#### 1.3.1.2 - Desenvolupament

En el desenvolupament del projecte s'han d'assolir els objectius del projecte ja definits. És per això que cada tasca ha de correspondre a un objectiu que s'ha d'acabar.

##### **Tasca 1**

T1.1. Comprovar que es pot accedir als SRPs de la base de dades de PABRE mitjançant els serveis web de PABRE (PABRE-WS).

T1.2. Entendre l'estructura d'un SRP i l'estructura dels documents que retornen els serveis web.

T1.3. Entendre els elements d'un SRP i com s'usen els SRP en la definició de requisits.

##### **Tasca 2**

T2.1. Estudiar la documentació per desenvolupar aplicacions de Chrome.

T2.2. Escollir el software més adequat pel desenvolupament de l'aplicació.

T2.3. Analitzar la millor manera d'estructurar l'aplicació a desenvolupar.

##### **Tasca 3**

T3.1. Desenvolupar una primera versió de PABRE-RW que permeti escriure i guardar requisits en una base de dades local.

##### **Tasca 4**

T4.1. Desenvolupar una segona versió de PABRE-RW que permeti recuperar els requisits d'una base de dades local, editar-los i guardar-los altre cop modificats.

**Tasca 5**

T5.1. Crear una tercera versió de PABRE-RW on s'afegeixi una nova àrea en la que es mostrin els SRPs i el contingut de cada un d'ells.

**Tasca 6**

T6.1. Crear una quarta versió de PABRE-RW on s'afegeixin facilitats de cerca/ordenació dels SRPs i del contingut de cada un d'ells.

**Tasca 7**

T7.1. Crear una cinquena versió de PABRE-RW on es pugui seleccionar una part d'un SRP i usar-la en la definició de requisits.

**Tasca 8**

T8.1. Crear una sisena versió de PABRE-RW on s'afegeixi la cerca d'un esquema (*Classifier Schema*), que és un sistema jeràrquic de classificació dels SRPs.

**Tasca 9**

T9.1. Crear una setena versió de PABRE-RW on es puguin seleccionar uns projectes ja predeterminats que poden servir d'exemple.

Les tasques del 1 al 7 han acabat abans del que s'havia previst, i això ha permès avançar i afegir dues tasques addicionals (la 8 i la 9). La planificació del desenvolupament ha canviat per aquest motiu.

## 1.3.1.3 Fase final

Finalment, en la darrera fase ja s'ha desenvolupat el projecte i ja es pot documentar i defensar. En realitat, s'ha començat a documentar abans d'acabar la fase de desenvolupament, ja que es podien anant escrivint parts dels capítols pertinents.

## 1.3.1.4 Hores totals

En la següent taula es mostra les hores dedicades de cada tasca:

Fases	Hores dedicades
Fase inicial	90
Desenvolupament	310
Tasca 1	40
Tasca 2	40
Tasca 3	50
Tasca 4	50
Tasca 5	50
Tasca 6	40
Tasca 7	40
Tasca 8	30
Tasca 9	30
Fase final	80
<b>Total</b>	<b>540</b>

*Taula 2: Hores dedicades*

### 1.3.1.5 Diagrama de Gantt

Les següents figures són els diagrames de Gantt que mostren la planificació temporal de cada fase.

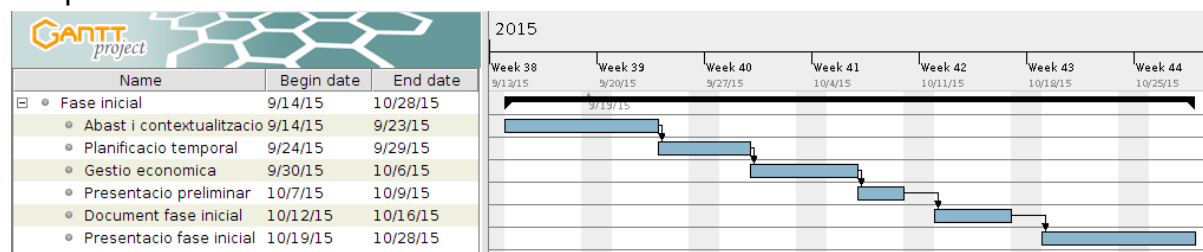


Figura 4: Fase inicial

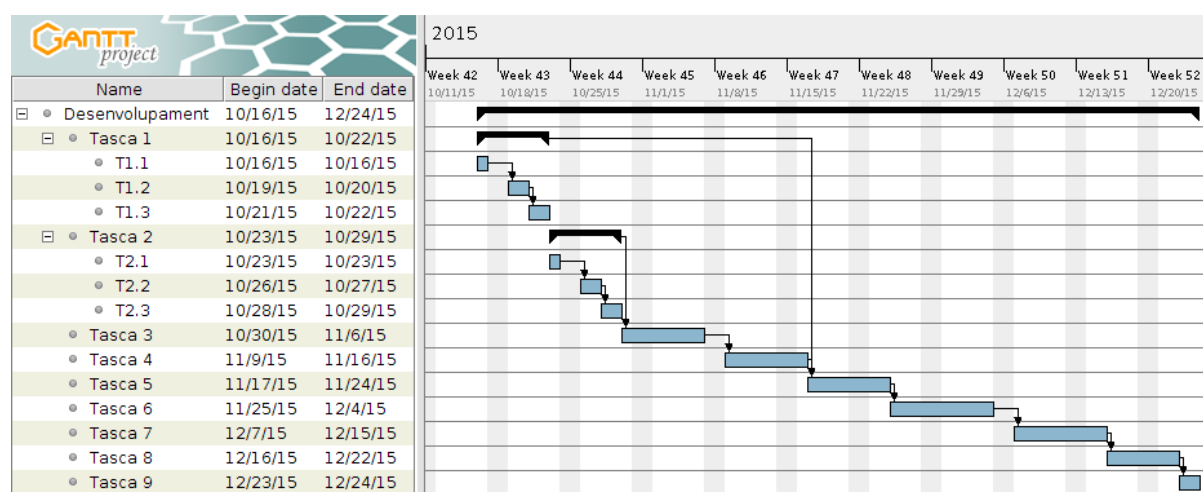


Figura 5: Desenvolupament

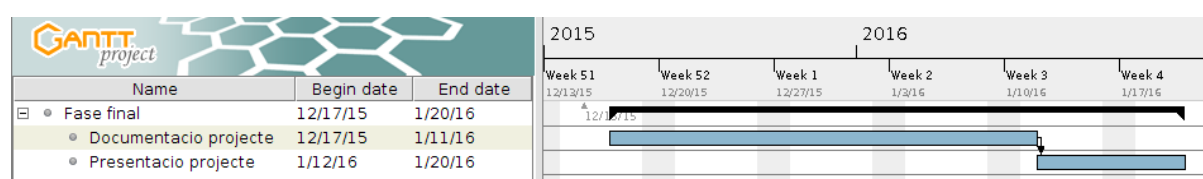


Figura 6: Fase final

## 1.3.2 - Valoració d'alternatives i pla d'accions

Com es pot veure, les tasques 1 i 2 es poden fer paral·lelament. La realització de la tasca 3 exigeix haver realitzat de la tasca 2 prèviament. També la tasca 4 exigeix haver realitzat la tasca 3.

La primera alternativa va ser doncs variar el temps de les tasques 1 i 2 intentant adaptar el temps de cadascuna a les necessitats que sorgissin. D'aquesta manera,

en cas d'existir problemes amb la tasca 1 sempre es podria continuar amb la tasca 2 i, mentrestant, anar resolent els possibles obstacles que podrien sorgir.

Un cop realitzades les tasques 1, 2, 3 i 4 ja es podria realitzar la 5. Les tasques 5, 6 i 7 s'han de realitzar seqüencialment. Un cop realitzades s'han realitzat les tasques 8 i 9, també seqüencialment.

En ser un projecte en una plataforma i llenguatge que no s'havia treballat anteriorment, no se sabia molt bé quina seria la complexitat de cada una de les tasques. Era possible que un cop s'hagués finalitzat la tasca 5, el projecte hagués arribat a l'abast del TFG. És per això que s'ha tingut en compte unes tasques bàsiques (que s'havien d'assolir) i unes altres tasques que depenien de la complexitat de les anteriors també assolir-les o no. Les tasques 1, 2, 3, 4 i 5 són tasques bàsiques. En canvi, les tasques 6 i 7 no ho són, aquestes tasques només es farien si hi hagués temps sobrer després de realitzar les bàsiques. Finalment s'ha assolit totes les tasques i ampliat a les tasques 8 i 9.

Per cada una de les tasques s'ha tingut en compte un temps de marge, en cas que fos necessari i es pogués allargar la tasca.

A més, s'ha tingut en compte un marge d'error de tot el projecte per tal d'acabar totes les tasques davant de qualsevol complicació no prevista i disposar de temps per documentar. Per, en cas que fos necessari, hi hagués temps per poder acabar totes les tasques i es tingués prou temps per documentar-lo.

S'han fet reunions setmanals amb la professora per comprovar que cada tasca s'assolia amb èxit.

## **1.4 – Gestió econòmica**

### **1.4.1 - Identificació i estimació de costos**

#### **1.4.1.1 Costos directes**

Els costos directes són aquells que es poden associar directament a la producció d'un sol producte, i que per tant només han de figurar en la comptabilitat de costos d'aquest producte. Els únics costos directes existents han sigut els sous dels recursos humans.

#### **Recursos humans**

Aquest no és un projecte d'empresa, sinó que es fa per a benefici d'un grup de recerca de la UPC, concretament el grup de recerca GESSI. L'única persona implicada en el desenvolupament sóc jo mateixa, que he fet tots els rols. Les altres

dues persones implicades, són membres del grup de recerca que han fet el paper de Clients. Per tant el seu temps no s'ha imputat com un cost.

La taula 4 mostra el cost dels recursos humans a partir dels rols de cada tasca mencionada en el diagrama de Gantt. La taula 3 mostra el cost de cada rol.

Rols	Preu per hora (€)
Cap de projecte (CP)	30
Analista / Dissenyador (A)	26
Tècnic programador (TP)	24
Tester (T)	24
Documentador (D)	28
Tècnic sistemes (TS)	24

Taula 3: Rols i els preus de cada rol.

Fases	CP	A	TP	T	D	TS	Hores dedicades	Cost (€)
Fase inicial	80	0	0	0	10	0	90	2680
Desenvolupament								
Tasca 1	34	3	0	0	3	0	40	1182
Tasca 2	1	20	0	0	1	18	40	1010
Tasca 3	1	8	35	5	1	0	50	1226
Tasca 4	1	8	35	5	1	0	50	1226
Tasca 5	1	8	35	5	1	0	50	1226
Tasca 6	1	10	50	8	1	0	40	1710
Tasca 7	1	10	50	8	1	0	40	1710
Tasca 8	1	10	50	8	1	0	30	1710
Tasca 9	1	10	50	8	1	0	30	1710
Fase final	5	0	0	0	75	0	80	2250
Total								17640

Taula 4: Pressupost recursos humans

#### 1.4.1.2 Costos indirectes

Els costos indirectes són els que afecten al procés productiu en general d'un o més productes, per la qual cosa no es poden assignar directament a un sol producte sense usar algun criteri d'assignació.

Els costos indirectes han estat: el hardware i software usats, el cost de l'electricitat consumida en el projecte, el cost de la connexió a internet usada durant el projecte, el cost de desplaçaments a les reunions amb el Client (la directora del projecte i la Cristina Palomares són les persones del grup GESSI que fan el paper de Client, i amb elles es fan les reunions setmanals).

#### Hardware

El hardware que s'ha usat pel desenvolupament del projecte no s'ha adquirit només pel projecte. Concretament s'ha utilitzat el meu l'ordinador de sobretaula:

	Eina	Preu	Vida útil	Temps de servei	Amortització total estimada
Hardware	Ordinador de taula	800€	4 anys	4 mesos	67€
	total				67€

Taula 5: Pressupost hardware

## Software

El software que s'ha usat per al desenvolupament ha estat software lliure i software sense cost per a projectes fets per estudiants o professors universitaris.

L'estimació de costos pel software és:

	Producte	Preu
Software	Ubuntu 12.04.5 LTS	-
	WebStorm	-
	Dart	-
	Chrome Apps & Extensions Developer Tool	-
	Pabre-ws API v1.0	-
	LibreOffice	-
	Adobe Acrobat Reader	-
	Google Drive	-
	Total	0€

Taula 6: Pressupost software

## Consum elèctric

El projecte s'ha desenvolupat a casa meva. El cost de l'energia consumida es calcula considerant que:

- les hores que s'han de dedicar al projecte 540 h
- el consum (ordinador+llum) és de 800 W
- per tant, l'energia gastada serà  $800 \text{ W} * 540 \text{ h} = 432 \text{ kWh}$
- el preu kWh és de 0,199338 €
- el cost serà per tant:  $432 \text{ kWh} * 0.199338 \text{ €/kWh} = 86.12 \text{ €}$

## Consum connexió a internet

El projecte s'ha desenvolupat a casa meva. La factura ADSL s'ha de repartir tenint en compte només el cost del consum del meu ordinador en les hores que es dediquen al projecte. Per simplificar s'ha decidit dividir el cost de la factura entre el número d'aparells que es connecten a l'ADSL i considerar el percentatge de temps d'utilització de l'ordinador durant el projecte:

- hi ha 7 aparells que es connecten a l'ADSL de casa
- el 90% de temps que s'utilitzi l'ordinador de sobretaula és el dedicat del projecte
- el preu de l'ADSL de casa és de 33 €/mes
- el cost serà per tant:  $33 \text{ €/mes} / 7 \text{ aparells} * 4 \text{ mesos} * 90\% = 16.97 \text{ €}$



## Desplaçaments

Per a assistir a les reunions setmanals s'ha usat transport públic, per tant cada setmana s'han fet dos viatges, un d'anada i un de tornada. S'han usat targetes T10:

- $16 \text{ reunions} * 2 \text{ viatges/reunió} = 32 \text{ viatges}$
- $32 \text{ viatges} * 9.95\text{€ per targeta} / 10 \text{ viatges} = 31,84\text{€}$

### 1.4.1.3 Consideració de riscos

Certs riscos podien augmentar el temps previst i podien augmentar el cost del projecte:

- Que jo em posés a treballar abans d'acabar el projecte i no tingués temps d'acabar-lo en el període previst: això era un risc minimitzable, ja que no tenia intenció de fer treball laboral abans d'acabar el projecte. En cas de rebre alguna oferta prou bona com per a no deixar-la passar, intentaria pactar començar el contracte al febrer i, si això no fos possible, començar amb un horari reduït fins al gener per deixar hores suficient pel projecte.
- Que sorgís algun problema amb les tecnologies que es pensava usar amb el que no fos factible implementar el projecte, o que no es tingués temps en un quadrimestre d'acabar la implementació: planificar molt bé la temporalització del projecte deixant marge per aquests imprevistos. A més, es va intentar tenir tecnologies de reserva en cas de que sorgís algun problema en alguna d'elles.
- Que em posés malalta durant el quadrimestre i que no tingués el temps suficient per acabar el projecte en el temps previst: necessitaria més temps per acabar el projecte.

En els tres casos una solució podia haver estat lliurat el projecte a l'abril, tot i que aquesta solució es reservava sempre com últim recurs.

Risc	Probabilitat	Coste unitari (€)	Cost (€)
treball laboral	4%	12000	480
tecnologies	20%	12000	2400
malaltia	5%	12000	600
<b>Total</b>			<b>3,480€</b>

*Taula 7: Costos de riscos*

En realitat no s'ha donat cap cas dels mencionats.

### 1.4.1.4 Consideració per imprevistos

Els imprevistos que podrien haver ocorregut en fer el projecte són tant avaries de hardware com que algun software que no s'ha tingut en compte d'entrada s'hagués de pagar per poder-lo fer servir:

- Avaries del hardware: En aquest cas s'utilitzaria una torre antiga que fa molt poc va ser substituïda per l'ordinador actual, per tant a cost 0€ excepte per les mínimes fraccions d'hora perdudes de fer el canvi.
- Productes software no lliures que s'han d'adquirir: com tot el projecte s'està fent i es pretenia fer amb software lliure o llicències gratuïtes per estudiants, la probabilitat d'haver d'adquirir un software propietari era molt petita.

Imprevist	Riel de que pass	Quantitat	Cost unitari (€)	Cost (€)
Averia	2%	1	0€	0€
Software necessari	1%	1	100€	1€
<b>Total</b>				<b>1€</b>

*Taula 8: Costos de imprevistos*

En aquest apartat tampoc no s'ha donat cap dels cassos considerats.

#### 1.4.1.5 Resum de costos

Com es pot veure al pressupost final, els recursos humans són els costos que més pugen. En software no necessitem cap pressupost ja que es totalment gratuït i en hardware utilitzem una quantitat molt petita. És per això que aquest projecte és viable.

Recursos humans	14,220€
Software	0€
Hardware	67€
Consum elèctric	86€
Consum connexió a internet	17€
Desplaçaments	32€
Riscos	3,480€
Imprevistos	1€
<b>Total</b>	<b>17,903€</b>

*Taula 9: Pressupost total*

### 1.4.2 - Control de gestió

Per comparar i avaluar las desviacions entre el pressupost plantejat i els costos reals del projecte s'ha de fer un seguiment i calcular els desviaments que hi pugui haver.

En el cas d'aquest projecte no s'ha produït cap desviació.

## 1.5 – Sostenibilitat i compromís social

La sostenibilitat d'un projecte s'ha de mirar des de diferents punts de vista: econòmic, social i ambiental.

### 1.5.1 - Econòmica

El projecte cal que condueixi a un producte que sigui durable al llarg del temps. A més cal que econòmicament siguin possible (viable), tant en la posada en marxa del producte com en el seu manteniment i continuïtat (durabilitat).

### 1.5.2 - Social

El vessant social de la sostenibilitat fa referència als recursos humans, tant des del punt de vista individual (ergonòmic, psicològic, ètic, etc.) com de comunitat (drets humans, valors socials, interacció a la societat, comunicació, decòrum, etc.).

L'aplicació que es desenvolupa en el projecte és una aplicació per ajudar als enginyers de requisits en la seva feina, i en concret fer la seva feina més fàcil en base a l'ús de patrons de requisits. D'aquesta manera, si la feina es fa més fàcil, entenem que si que oferirà aquest aspecte de sostenibilitat social.

### 1.5.3 - Ambiental

Com a producte purament de software on no es fabrica res, ni es treballa amb res material, l'únic factor que podem considerar és la despesa d'energia utilitzada en el procés de desenvolupament i en el cicle de vida del producte. Forçant conceptes podríem considerar l'impacte de la construcció dels ordinadors usats, però això queda fora del nostre abast ja que ignorem el procés usat a la fabricació i no el podem esbrinar.

Respecte de l'energia consumida en el procés de desenvolupament hem de considerar un sol PC d'un consum d'uns 800W durant 540 hores que fa un consum total de  $540 \times 800 = 432$  kWh.

L'energia consumida durant l'explotació de l'aplicació requerirà més aparells ja que pot afectar a més d'un servidor. Donat que l'ordinador que allotjarà el servidor no es dedicarà exclusivament a la nostra aplicació cal considerar la fracció de temps que la CPU es dedica a ella. Això fa molt difícil el càlcul ja que ignorem el percentatge de servidor usat. De totes maneres, probablement l'ordinador que allotja al nostre servidor funcionaria igualment si no la poséssim, així que en aquest aspecte podem considerar que no hi ha cost ambiental afegit.

Pel que fa l'impacte ambiental dels 432 kWh tampoc ho podem especificar perquè depèn de com el proveïdor produeix aquesta energia elèctrica. Però en qualsevol cas, l'impacte serà mínim.

### 1.5.4 - Taula de sostenibilitat

Sostenibilitat	Econòmica	Social	Ambiental
Planificació	Apartat 5.1	Apartat 5.2	Apartat 5.3
Valoració	6	6	6

*Taula 10: Sostenibilitat*

## 1.6 – Estructura del document

Després de la introducció, aquest document està estructurat per tasques. Cada tasca que hi ha en la planificació del desenvolupament correspon a un capítol.

En cada capítol explicaré la feina que he anat fent, a més dels problemes que hagi pogut tenir i com s'han pogut solucionar. També explicaré com s'ha anat desenvolupant la interfície i la seva usabilitat.

Un cop finalitzats aquests capítols, he afegit les conclusions del projecte, el glossari i els annexos. Als annexos hi ha l'índex de taules, l'índex de figures, el manual d'instal·lació i el manual d'usuari.

## 2 – Estudi del domini

En aquest capítol s'explica com s'ha assolit la primera tasca corresponent l'objectiu 1 (explicat en el capítol 1 secció “**Abast del projecte**”). La tasca consistia en:

- Entendre l'estructura d'un SRP i l'estructura dels documents que retornen els serveis web.
- Comprovar que es pot accedir als SRPs de la base de dades de PABRE mitjançant els serveis web de PABRE (PABRE-WS).

### 2.1 – Estructura dels SRPs

En aquesta secció del capítol explicaré l'estructura dels SRPs. Aquesta estructura està emmagatzemada al PABRE-WS. Explicaré com està emmagatzemada l'estructura d'un SRP. També explicaré com està emmagatzemada l'estructura de tots els patrons i com es poden classificar.

La següent figura mostra la estructura bàsica de les dades que treballarem:

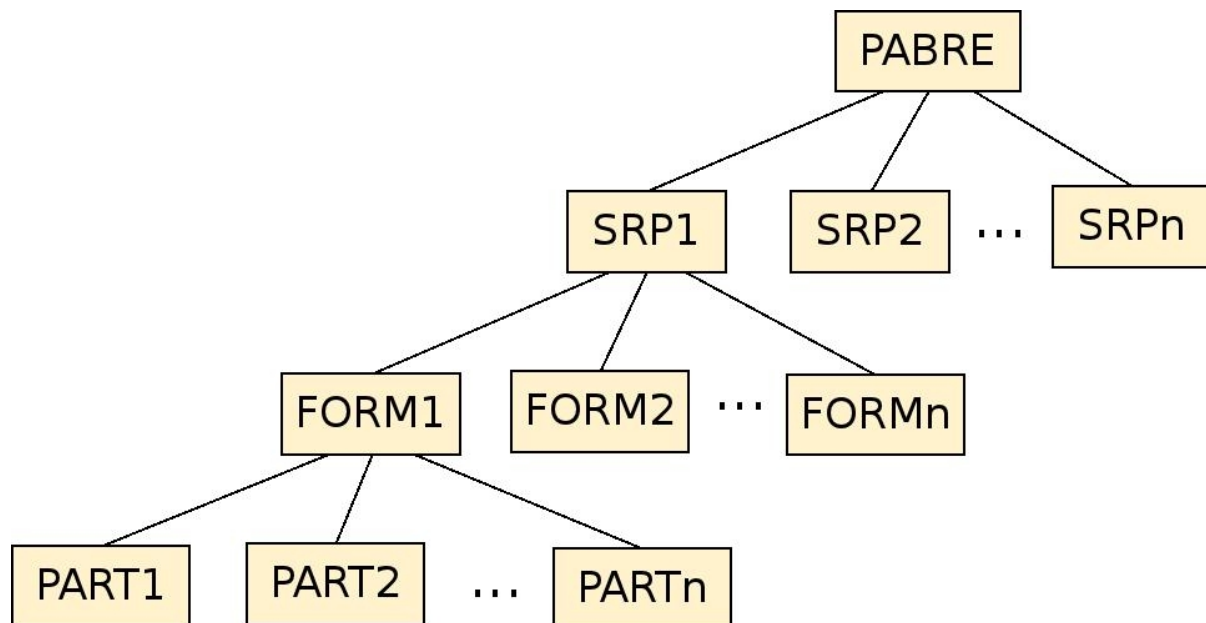


Figura 7: Estructura bàsica dels SRPs

#### 2.1.1 - Estructura d'un Patró de Requisits

Cada *patró* (*pattern*) correspon a un ventall de requisits relacionats, que queden definits dins del marc de l'*objectiu* del patró (*pattern goal*).

Cada patró incorpora un o més formularis (*forms*).

Tot seguit posarem un exemple del qual mostrem aquí l'estructura bàsica:

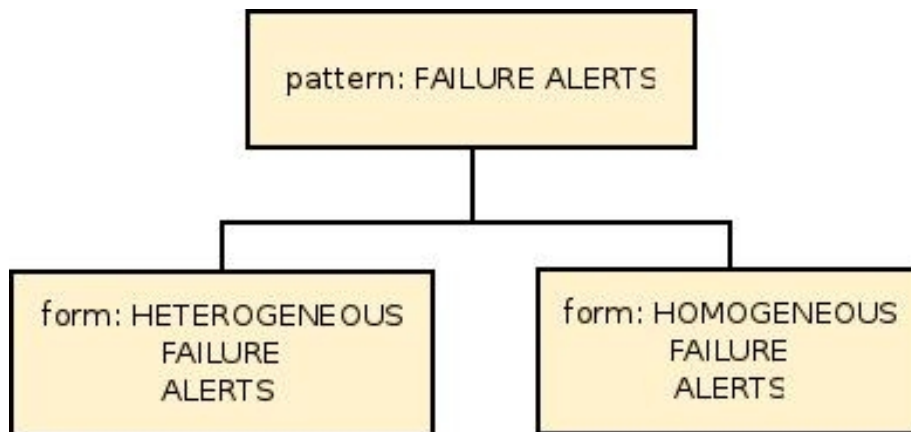


Figura 8: estructura bàsica del patró Failure Alerts

en la qual veiem que aquest patró té dos forms.

Cada form, al seu torn està compostat per diverses *parts*, que són les que contenen els *requisits*.

A cada form sempre hi ha d'haver una part anomenada part fixe (*fixed*), i hi pot haver altres parts anomenades esteses (*extended*). Tots els forms dels patrons de PABRE tenen sempre una part fixa.

La taula 11 mostra el patró sencer de *Failure Alerts*:

<b>Requirement Pattern Failure Alerts</b>	<b>Description</b>	This pattern expresses the need of having the system functionality to inform users about system failures at the moment the failure occurs.
	<b>Comments</b>	-
	<b>Pattern goal</b>	Alert users about system failures
	<b>Author</b>	GESSI-CITI
	<b>Sources(0...*)</b>	<ul style="list-style-type: none"> <li>Requirement books from CITI</li> <li>Specialized literature</li> </ul>
	<b>Keywords (0...*)</b>	["technical support", "Alert", "crash response", "Failure", "System failure"]
	<b>Dependencies (0...*)</b>	-

<b>Requirement Form Heterogeneous Failure Alerts</b>	<b>Description</b>	This form establishes a dependency among the type of alert and the type of system failure that occurs. The extension establishes which alerts are issued by which system failures	
	<b>Comments</b>	Application of extensions:\tAlerts for Failure Type: may (usually will) be applied more than once	
	<b>Modification Date</b>	2009-01-27T23:00:00.000+0000	
	<b>Author</b>	GESSI-CITI	
	<b>Sources(0...*)</b>	<ul style="list-style-type: none"> <li>Requirement books from CITI</li> <li>Specialized literature</li> </ul>	
	<b>Fixed</b>	<b>Template</b>	The system shall trigger different types of alerts depending on the type of failure
		<b>Extended Parts Constraint</b>	multiplicity(Alerts for Failure Types) = 0..*
	<b>Extended Part Alerts for Failure Types</b>	<b>Template</b>	The system shall trigger %alerts% alerts in case of %failures% failures
		<b>Parameter</b>	<b>Metric</b>
		alerts: non-empty set of alert types	alerts: Set(AlertType) AlertType: Domain of possible types of alerts
		failures: non-empty set of failure types	failures: Set(FailureType) FailureType: Domain of possible types of failures
<b>Requirement Form Homogeneous Failure Alerts</b>	<b>Description</b>	This form does not establish any relationship among the type of alert and the type of system failure. It has extensions to determine the type of system failures and alerts in the system	
	<b>Comments</b>	Application of extensions:\tAlert Types, Failure Types: may be applied at most once each.	
	<b>Modification Date</b>	2009-03-19T23:00:00.000+0000	
	<b>Author</b>	GESSI-CITI	
	<b>Sources(0...*)</b>	<ul style="list-style-type: none"> <li>Requirement books from CITI</li> <li>Specialized literature</li> </ul>	

	<b>Fixed</b>	<b>Template</b>	The system shall trigger an alert in case of failure.
		<b>Extended Parts Constraint</b>	multiplicity(AlertsTypes) = 0..1 and multiplicity(Failure Types) = 0..1
	<b>Extended Part Alerts Types</b>	<b>Template</b>	The solution shall trigger %alerts % alerts in case of failure
		<b>Parameter</b>	<b>Metric</b>
		alerts: non-empty set of alert types	alerts: Set(AlertType) AlertType: Domain of possible types of alerts
	<b>Extended Part Failure Types</b>	<b>Template</b>	The system shall trigger alerts in case of %failures% failures
		<b>Parameter</b>	<b>Metric</b>
		failures: non-empty set of failure types	failures: Set(FailureType) FailureType: Domain of possible types of failure

Taula 11: patró Failure Alerts

Els primers atributs (**Requirement Pattern Failure Alerts**) contenen el Pattern metadata, informació bàsica d'un SRP.

Aquest primer conjunt d'atributs (Pattern metadata) conté el nom del patró a la definició (**Failure Alerts**) i diversos camps amb les dades del SRP:

- **Description:** la descripció del SRP
- **Comments:** comentaris afegits per l'autor i/o pels usuaris (cap en aquest cas).
- **Pattern goal:** l'objectiu del patró de requisits.
- **Author:** l'autor, que pot ser personal o institucional com és el cas aquí
- **Sources:** les fonts d'informació d'on s'ha obtingut
- **Keywords:** algunes paraules clau per ajudar en la recerca en el repositori.
- **Dependencies:** permeten que els patrons de requisits es puguin relacionar entre ells. En l'exemple no hi ha cap dependència amb cap altre patró.

Aquests són els atributs bàsics per a tots els patrons. Després apareixen els formularis (forms). Cada patró divideix la informació en un o més forms: d'aquesta manera es pot separar la informació necessària per granularitat.

En cada un dels formularis hi ha diversos atributs. Els primers atributs d'un formulari són similars als atributs bàsics d'un patró: nom del formulari, la seva descripció, el



seu autor, comentaris afegits per l'autor i pels usuaris, la data en la que s'han canviat per últim cop i les fonts d'informació d'on s'han obtingut.

Ara, a més dels camps anteriors, en cada form hi ha una única part **fixa** i pot tenir opcionalment parts **esteses**.

Aquestes dues parts corresponen a diversos refinaments del patró, defineixen i concreten les accions o mètodes que portaran a complir l'objectiu global.

Com es pot veure, el patró **Failure Alerts** conté dos formularis i en cada un d'ells conté les fixed parts i les extended parts.

- **Heterogeneous Failure Alert.** Conté les següents parts:
  - Una fixed part
  - Una extended part: **Alerts for Failure Types**
- **Homogeneous Failure Alert.** Conté les següents parts:
  - Una fixed part
  - Dues extended parts: **Alerts Types** i **Failure Types**.

Cada part fixa (*fixed part*) té un *form text* (explicat més endavant). La *fixed part* ha d'aconseguir que l'objectiu (definit en *pattern metadata*) es compleixi.

En el form *Heterogeneous Failure Alert* hi ha una *fixed part* amb la seva descripció (*templated*) i quines extended parts conté, que en aquest cas és *Alert for Failure Types*. En aquest exemple no hi ha cap mètrica ja que no hi ha cap variable. Els '%' són per posar entre ells les variables, definides en les mètriques.

Cada extensió (*extended part*) té el seu nom i un *form text* (explicat més a baix). Les *extended part* ajuden a ampliar la informació sobre l'objectiu ja assolit per les *fixed parts*. En l'exemple hi ha l'*extended part* d'*Alert for Failure Types*, del form *Heterogeneous Failure Alert*. Té el nom, descripció, etc. i a més a més té les mètriques.

Cada part fixa i estesa d'un patró és especificat per un form text. Aquest text s'expressa com una frase escrita en llenguatge natural que pot incloure un o més paràmetres indicats per les parts que poden variar en diferents projectes. Per a definir els valors vàlids que un paràmetre pot prendre, cada paràmetre s'uneix a una mètrica.

Les mètriques poden enumerar valors (per exemple, noms de plataformes middleware), nombres sencers (per exemple, per indicant el nombre de connexions amb el suport), nombres reals (per exemple, per al mesurament de temps de resposta) i valors booleans (per exemple, per saber si algun protocol és compatible). Com es pot veure en l'exemple *Alert for Failure Types*, del form *Heterogeneous* hi ha dues variables:

- *alerts*: els possibles valors són els indicats dintre de la funció `set(Bip, Mail, SMS, ...)`
- *failures*: els possibles valors són els indicats dintre de la funció `set(network crash, database crash, ...)`.

### 2.1.2 - Patrons i esquemes

L'agrupació de requisits mitjançant els patterns significa un notable grau d'abstracció, però pot ser insuficient.

Per facilitar-ne la reutilització i la comprensió és convenient que estiguin indexats segons un esquema de classificació jeràrquica.

Aquest sistema de classificació jeràrquica són els esquemes (classifier schemas) que són per facilitar la comprensió i la reutilització durant el procés d'obtenció, els patrons s'ordenen en uns esquemes jeràrquics de classificació.

Existeixen diversos esquemes diferents per accedir als SRPs. Cada esquema reflexa un sistema de classificació diferent dels altres.

Bàsicament un esquema és un arbre en el qual les fulles són els SRPs. Els nivells de l'arbre corresponen a conceptes més afinats i menys genèrics conforme es baixa de nivell.

Al PABRE-WS es podrà accedir als SRPs mitjançant diversos esquemes diferents:

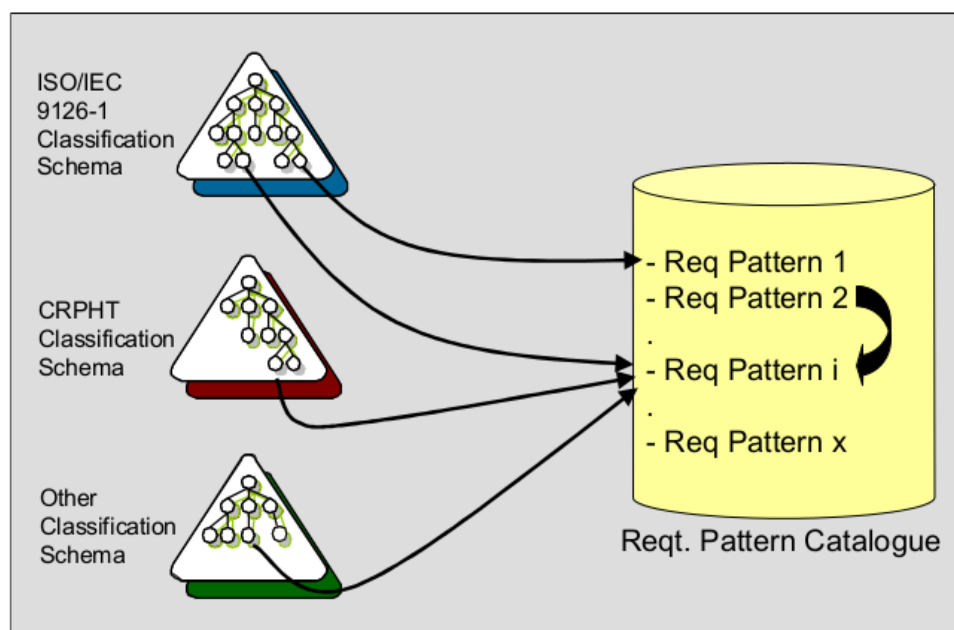


Figura 9: Esquemes del sistema PABRE

### 2.1.3 - Model conceptual de l'estructura

Aquesta estructura es pot veure representada en UML en la figura 10

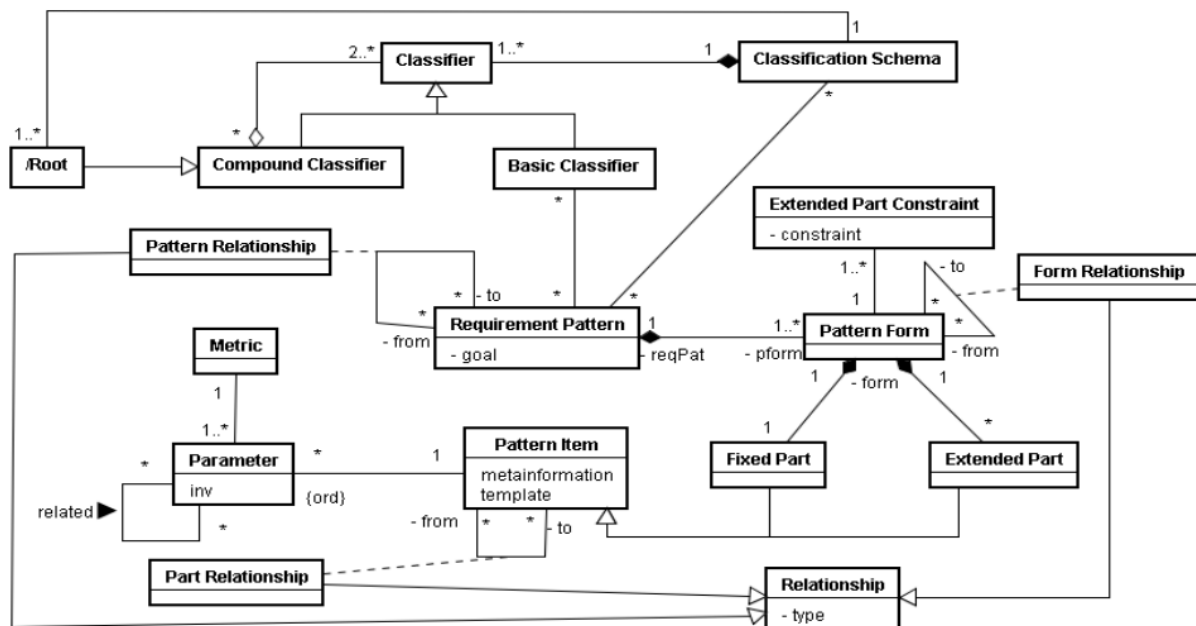


Figura 10: Diagrama UML

Com es pot veure, a la figura 9 es veu el UML de l'estructura del catàleg de PABRE. En aquest catàleg hi ha dues classes principals:

- *Classification schemes*: classe principal per la classificació de esquemes.
- *Requirement pattern*: classe principal que conté el patró (SRP) i la seva informació.

En primer lloc hi ha la classificació d'esquemes (*Classification Schema*). És l'arrel d'on estan classificats els patrons. Aquesta classificació va a una classe jeràrquica, *Classifier*, per saber si la composició és composta o no.

Si és composta s'ha de seguir baixant de nivell; en cas contrari, arribem a la classificació bàsica que és on hi ha el patró, *Requirement Pattern*. Aquest *Classification Schemas* configura un arbre jeràrquic en el qual els patrons són elements terminals.

La classe *Requirement Pattern* conté el patró i la seva informació. Cada patró ha d'estar com a mínim relacionat amb un form.

Cada form (*Pattern Form*) ha d'estar descomposat en parts. Ha de tenir una única *Fixed part*, i opcionalment pot tenir diverses *Extended Parts*. Cada una d'elles, de les fixed i extended parts, contenen la mateixa estructura d'informació. També pot contenir informació variable, opcionalment, en *Parameter*. El *Parameter* estableix la *Metric*. Diversos *Parameter* poden contenir la mateixa *Metric*.

## 2.2 – Accés als patrons de requisits via PABRE-WS

El sistema PABRE conté un subsistema anomenat PABRE-WS. Aquest subsistema consisteix en serveis web REST que permeten a altres sistemes externs a PABRE accedir a la base de dades de SRP.

Un servei web REST (Representational State Transfer) [20] és un estil d'arquitectura de programari que es fa servir en aplicacions de xarxa distribuïdes, típicament sobre HTTP com és el nostre cas. REST es un conjunt de normes com: tenir interfícies uniformes, ser *stateless*, tenir el client i el servidor separats, etc.

Les restriccions de REST estan pensades per poder tenir un sistema que sigui eficient, escalable, amb interfícies simples, fàcil de modificar, portable i fiable.

PABRE-WS està allotjat en un servidor de la UPC. Tota la documentació està en l'API [23]. Aquesta API explica com es pot accedir a PABRE-WS des de qualsevol ordinador, i com està estructurat. A la següent figura es pot veure com està l'adreçament de PABRE-WS:



Figura 11: Esquema de l'API [21]

L'estructura de l'API (PABRE-WS) té les següents adreces:

- *Patterns collection* **/api/patterns**: conté una llista de tots els patrons que hi ha al sistema i ordenada alfabèticament. Cada un dels patrons llistats té unes propietats i una URI (Uniform Resource Identifier) que va a un Individual Pattern per accedir a la resta de la informació.
- *Individual Pattern* **/api/patterns/{patternid}**: conté tota la informació del patró, es sol·licita amb la URI del patró que hi havia en Patterns collection i retorna els forms, les fixed i extended parts, etc. Si no s'explicita la versió, retorna la última de totes.
- *Pattern versions collection* **/api/patterns/{patternid}/versions**: aquesta direcció conté totes les versions d'un patró. Té un conjunt d'atributs però per tenir la informació completa s'ha de accedir a la URI de la versió que es vol conèixer, és a dir, de Individual Pattern version.
- *Individual Pattern version* **/api/patterns/{patternid}/versions/{versionid}**: aquest recurs conté la informació d'un patró d'una versió concreta.
- *Metrics collection* **/api/metrics**: conté la llista de totes les mètriques del sistema. Conté un conjunt de informació de cada mètrica però per tenir tota la informació s'ha d'accedir a l'URI de cada una d'elles, que està en *Individual Metric*.
- *Individual Metric* **/api/metrics/{metricid}**: conté tota la informació d'una mètrica concreta.
- *Sources collection* **/api/sources**: conté la llista de totes els sources (fonts) del sistema. Conté un conjunt d'informació de cada source però per tenir tota la informació s'ha d'accedir a l'URI de cada un d'ells, en *Individual Source*.
- *Individual Source* **/api/sources/{sourcesid}**: conté tota la informació d'un source concret.
- *Schemas collection* **/api/schemas**: conté una llista de tots els esquemes que hi ha al sistema. Cada un dels esquemes té unes propietats i té una URI que va a un esquema individual per accedir a la resta de la informació.
- *Individual Schema* **/api/schemas/{schemaid}/classifiers**: representa un esquema individual. Aquest esquema conté una jerarquia que conté els classificadors.
- *Classifiers collection* **/api/schemas/{schemaid}/classifiers/{classifierid}**: aquest recurs retorna la jerarquia d'un arbre d'un classificador específic. Conté un conjunt d'atributs limitats, però per accedir a la resta es farà a través de la URI que anirà a *Individual Classifier*.
- *Individual Classifier*: representa la informació que té un classificador individual, conté les URIs dels seus patrons.

Per al meu projecte ha calgut obtenir la informació dels patrons a partir de l'esquema de l'API [22]. És a dir, ha calgut accedir a *Patterns collection*, que conté

una llista on cada element correspon a un patró. Cada element de la llista conté els següents camps:

- name
- available
- editable
- uri

D'aquesta manera és mostra una llista amb els patrons disponibles i editables. Mitjançant el valor contingut en el camp *uri* es pot accedir posteriorment a l'estructura del patró mencionada anteriorment i que conté tota la informació.

A l'inici d'aquesta tasca va ser necessari entendre com s'accedien a les webs des de *Dart*, llenguatge desenvolupat per Google (veure capítol 3). Per això es va utilitzar la classe `HttpRequest` [24]. Per accedir al servei web principal es faria així:

```
var path = 'http://pabre.essi.upc.edu/ws/api/patterns';  
libhtml.HttpRequest.getString(path).then(processPattern);
```

`path` és la variable que conté la direcció de PABRE-WS. Després amb la classe `HttpRequest` s'accedeix a la web i s'obté la informació json (subconjunt de Javascript que no requereix XML). Un cop obtinguda es descodifica amb la funció `processPattern` per obtenir un objecte de Dart.

Per cada un dels patrons s'ha hagut d'accedir a cada *uri* que hi havia i processar-la amb la classe `HttpRequest`. Un cop s'accedeix a l'*uri* s'ha de descodificar el JSON, igual que al primer exemple.

Aquest va ser el pas inicial. A més a més, s'ha necessitat entendre tota l'estructura (explicada a la secció “Estructura dels SRPs” del capítol 2) i conèixer la informació que era necessària. Per això es va necessitar accedir a tota la informació del patró, és a dir a *Individual Pattern*. En aquest cas es necessitava:

- noms del patrons i descripció
- noms de les forms i descripció
- descripció de les *Fixed Part*
- nom i descripció de les *Extended Part*.
- Els valors que tenen les mètriques de les fixed i extended parts.

També s'ha necessitat accedir a:

- Schemas collection
- Individual Schema

S'ha fet utilitzant el mateix mètode que accedint als patrons, però en aquest cas per accedir a un esquema específic.

## 2.3 – Model de casos d'ús

### 2.3.1 - Diagrama dels casos d'ús

El sistema PABRE-RW tindrà uns casos d'ús que seran els següents:

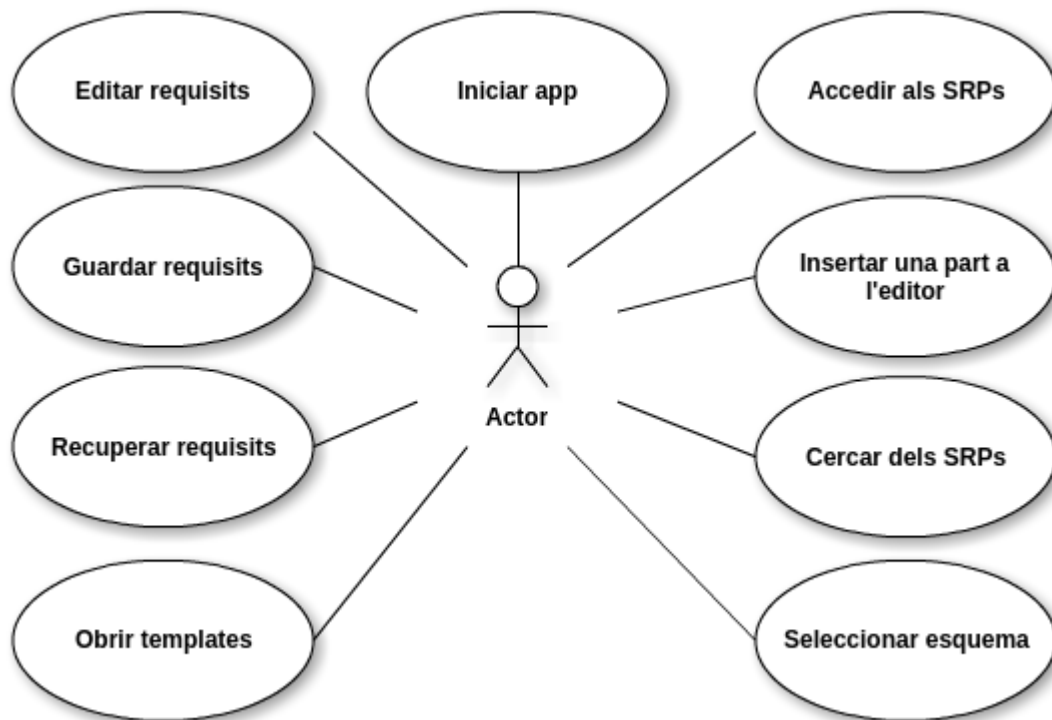


Figura 12: Casos d'ús

### 2.3.2 - Cas d'ús: Iniciar app

#### Descripció

L'usuari vol iniciar l'app.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol executar l'app.

**Escenari d'èxit principal:**

1. L'usuari inicialitza l'app.
2. El sistema mostra els noms dels patrons i els noms dels classificadors.
3. S'acaba el cas d'ús.

**Extensions:** -

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció "1 – Cas d'ús: Iniciar app" de l'**Annex: Anàlisi**.

### 2.3.3 - Cas d'ús: Editar requisits

#### Descripció

L'usuari vol poder escriure requisits i poder editar-los. A través d'aquest cas d'ús l'usuari podrà modificar els seus requisits.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol editar els requisits

**Escenari d'èxit principal:**

1. L'usuari gestiona la informació en l'editor de text.
2. S'acaba el cas d'ús.

**Extensions:**

- 1a. L'usuari usa l'editor de text.
- 1b. L'usuari vol canviar la negreta
  - 1b1. L'usuari selecciona la negreta
  - 1b2 El sistema activa/desactiva la negreta en l'editor
- 1c. L'usuari vol canviar la cursiva
  - 1c1. L'usuari selecciona la cursiva
  - 1c2 El sistema activa/desactiva la cursiva en l'editor
- 1d. L'usuari vol canviar el subratllat
  - 1d1. L'usuari selecciona el subratllat
  - 1d2 El sistema activa/desactiva el subratllat en l'editor
- 1e. L'usuari vol canviar el ratllat
  - 1e1. L'usuari selecciona el ratllat
  - 1e2 El sistema activa/desactiva el ratllat en l'editor
- 1f. L'usuari vol canviar la llista de punts
  - 1f1. L'usuari selecciona la llista de punts
  - 1f2 El sistema activa/desactiva la llista de punts en l'editor
- 1g. L'usuari vol canviar la llista de números
  - 1g1. L'usuari selecciona la llista de números
  - 1g2 El sistema activa/desactiva la llista de números en l'editor
- 1h. L'usuari vol canviar la indentació
  - 1h1. L'usuari selecciona la indentació
  - 1h2 El sistema afegeix una indentació en l'editor
- 1i. L'usuari vol canviar la desindentació
  - 1i1. L'usuari selecciona la desindentació
  - 1i2 El sistema afegeix una desindentació en l'editor
- 1j. L'usuari vol anar cap enrere
  - 1j1. L'usuari selecciona l'enrere
  - 1j2 El sistema canvia la informació de l'editor per la que hi havia anteriorment



- 1k. L'usuari vol anar cap endavant
  - 1k1. L'usuari selecciona l'endavant
  - 1k2 El sistema canvia la informació de l'editor per la que hi havia posteriorment.

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció “2 – Cas d'ús: Editar requisits” de l'**Annex: Anàlisi**.

### 2.3.4 - Cas d'ús: Guardar requisits

#### Descripció

L'usuari vol poder guardar els requisits que hi ha a l'editor de l'aplicació, en un fitxer en la base de dades local.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol guardar un fitxer en la base de dades local.

**Escenari d'èxit principal:**

- 1. L'usuari indica que vol guardar les seves dades.
- 2. S'ha acabat el cas d'ús.
- 3. El sistema obre un pop-up (una nova finestra) i mostra un missatge indicant que el fitxer s'ha guardat correctament

**Extensions:**

- 1a. L'usuari escriu el nom amb el que vol guardar el fitxer.
- 2a. L'extensió és errònia.
  - 2a1. El sistema mostra un pop-up i mostra un missatge amb l'extensió que ha de tenir el fitxer.
  - 2a2. L'usuari ha de prémer el botó per tancar el pop-up
  - 2a3. Es torna al punt 1.

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció “3 – Cas d'ús: Guardar requisits” de l'**Annex: Anàlisi**.

### 2.3.5 - Cas d'ús: Recuperar requisits

#### Descripció

L'usuari vol poder pujar un fitxer i mostrar-lo en l'editor de l'aplicació. Aquest fitxer està guardat a la base de dades local.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol pujar un fitxer a l'aplicació.

**Escenari d'èxit principal:**

1. L'usuari indica que vol pujar un fitxer.
2. El sistema obre una finestra on mostra els fitxers que té l'usuari en el seu ordinador.
3. L'usuari navega per les carpetes de la seva base de dades local fins a trobar el fitxer que vol.
4. L'usuari selecciona el fitxer que vol pujar.
5. El sistema mostra la informació que conté aquest fitxer.
6. S'acaba el cas d'ús.

**Extensions:**

- 1a. El sistema mostra una finestra amb un avís de confirmació.
  - 1a1. L'usuari ho confirma.
- 3a. L'usuari modifica el nom de les extensions que apareixen en els fitxers.
- 4a. L'extensió del fitxer és erroni
  - 4a1. El sistema mostra un pop-up i mostra un missatge amb l'extensió que ha de tenir el fitxer
  - 4a2. L'usuari ha de prémer el botó per treure el pop-up
  - 4a3. Es torna al punt 1.

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció “4 – Cas d'ús: Recuperar requisits” de l'**Annex: Anàlisi**.

### 2.3.6 - Cas d'ús: Accedir als SRPs

**Descripció**

L'usuari vol poder veure tota la informació que conté un patró, és a dir, la descripció del patró, els form, la descripció de cada form, les fixed i extended part.

**Especificació**

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari poder veure tota la informació que conté un patró.

**Escenari d'èxit principal:**

1. El sistema mostra els noms dels patrons.
2. L'usuari selecciona el patró que vol.
3. El sistema mostra informació del patró.
4. L'usuari selecciona una form.
5. El sistema mostra informació de la form seleccionada.
6. S'activa el cas d'ús “Insertar una part a l'editor”
7. S'acaba el cas d'ús.

**Extensions:**

-

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció “5 – Cas d'ús: Accedir als SRPs” de l'**Annex: Anàlisi**.

### 2.3.7 - Cas d'ús: Insertar una part a l'editor

#### Descripció

L'usuari vol poder escriure una part d'un patró a l'editor de text.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol poder escriure una part d'un patró a l'editor de text.

**Escenari d'èxit principal:**

1. L'usuari selecciona la part d'un patró
2. El sistema mostra la informació de les mètriques.
3. El sistema afegeix la part del patró a l'editor de text
4. S'acaba el cas d'ús.

**Extensions:**

- 1a. L'usuari vol veure les mètriques
  - 1a1 L'usuari selecciona un part
  - 1a2. El sistema mostra les mètriques
- 1b. L'usuari vol afegir una part a l'editor
  - 1b1 L'usuari selecciona el botó per afegir una part a l'editor
  - 1b2. El sistema mostra la part, el requisit, afegit a l'editor
  - 1b3. El sistema mostra les mètriques

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció "6 – Cas d'ús: Insertar una part a l'editor" de l'**Annex: Anàlisi**.

### 2.3.8 - Cas d'ús: Cercar els SRPs

#### Descripció

L'usuari ha de poder fer cerques de paraules per poder trobar amb aquestes paraules clau, els patrons que contenen aquestes paraules.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol fer recerca de paraules.

**Escenari d'èxit principal:**

1. L'usuari gestiona la cerca/ordenació dels SRPs.
2. S'acaba el cas d'ús.

**Extensions:**

- 1a. L'usuari vol cercar una paraula
  - 1a1. L'usuari escriu una paraula o paraules
  - 1a2. L'usuari cerca la paraula o paraules.
  - 1b3. El sistema mostra els SRPs que té aquest contingut.
- 1b. L'usuari vol netejar la recerca.

- 1b1 L'usuari selecciona que vol netejar la recerca
- 1b2 El sistema mostra tots els SRPs

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció “7 – Cas d'ús: Cercar els SRPs” de l'**Annex: Anàlisi**.

### 2.3.9 - Cas d'ús: Seleccionar esquema

#### Descripció

L'usuari ha de poder seleccionar una opció de l'esquema que es vol visualitzar.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol fer recerca de paraules.

**Escenari d'èxit principal:**

1. L'usuari selecciona una opció de l'esquema.
2. El sistema mostra els SRPs d'aquella opció
3. S'acaba el cas d'ús.

**Extensions:**

-

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció “8 – Cas d'ús: Seleccionar esquema” de l'**Annex: Anàlisi**.

### 2.3.10 - Cas d'ús: Obrir templates

#### Descripció

L'usuari ha de poder obrir projectes prefabricats de l'aplicació

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol visualitzar projectes ja predefinitos

**Escenari d'èxit principal:**

1. L'usuari selecciona el projecte que vol
2. S'acaba el cas d'ús.

**Extensions:**

- 1a. El sistema mostra una finestra amb un avís de confirmació.
  - 1a1. L'usuari ho confirma.
  - 1a2. El sistema mostra la informació en l'editor de text.

Els diagrames de seqüència, els contractes i les operacions d'aquest cas d'ús el podeu trobat a la secció “9 – Cas d'ús: Obrir templates” de l'**Annex: Anàlisi**.

## 3 – Estudi de la tecnologia

En aquest capítol s'explica com s'ha assolit la segona tasca corresponent a l'objectiu 2 (explicat en el capítol 1, secció “**Abast del projecte**”); per això s'ha hagut de:

- Estudiar la documentació sobre com desenvolupar aplicacions de Chrome.
- Escollir el software més adequat per al desenvolupament de l'aplicació.
- Analitzar la millor manera d'estructurar l'aplicació que es vol desenvolupar.

### 3.1 – Estudi per desenvolupar Apps de Chrome

Les Apps de Chrome són aplicacions d'escriptori en el sentit que funcionen localment i poden estar en una finestra pròpia. Cal, però, tenir el Chrome instal·lat i funcionant com a procés actiu, ja que l'App en depèn.

Com l'App està situada sobre el Chrome, aquest navegador actua com a nivell aïllant i independitza l'App de la plataforma sobre la que està corrent, de manera que les Apps són sempre multiplataforma.

Gràcies a que utilitzen HTML, CSS i JavaScript, només cal que es tingui el navegador de Chrome per poder baixar l'aplicació des de qualsevol ordinador i qualsevol sistema operatiu.

Tot això permet que una App de Chrome es pugui oferir a un públic molt més ampli.

Per a desenvolupar aplicacions de Chrome [25] es necessiten els següents fitxers:

- El *manifest.json* conté informació per obrir l'aplicació, el què és, com posar-la en marxa i els permisos addicionals que requereix.
- El *background.json* s'utilitza per crear la pàgina/finestra que contindrà el HTML de l'aplicació, i és l'esdeveniment responsable de la gestió del cicle de vida de l'aplicació..
- Tot el codi ha de ser inclòs en el paquet de Chrome App. Això inclou HTML, JS i CSS.
- Totes les icones i altres actius s'han d'incloure en el paquet també.

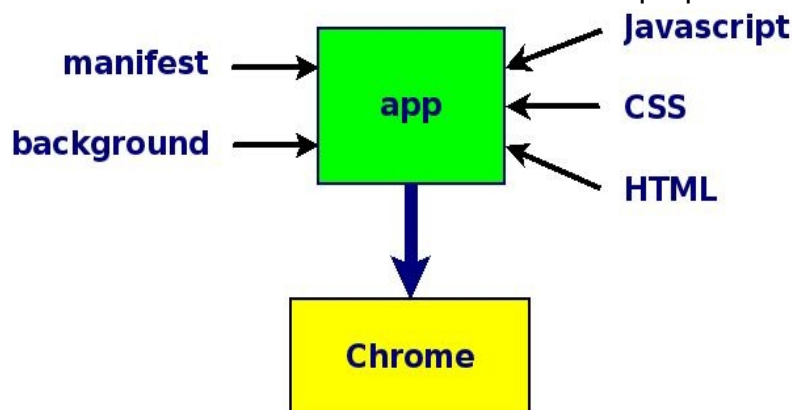


Figura 13: app de Chrome

**Manifest:**

Per cada aplicació de Chrome hi ha un arxiu manifest amb informació important. En el nostre projecte la informació important és:

- name: nom de l'aplicació
- descripton: descripció de l'aplicació
- version: la versió que té l'aplicació
- manifest\_version: és la versió del manifest. És important que la versió del manifest sigui 2, ja que la 1 està obsoleta a partir del Chrome 18 i s'està millorant per a la següent versió.
- Apps: aquest camp serveix per especificar els script que hi ha a l'aplicació.
- Icons: en aquest camp es posen els noms dels fitxers corresponents a les icones pertinents de l'aplicació.

```
{
  "short_name": "PABRE-RW",
  "name": "PABRE Requirements Specification Editor",
  "version": "0.4",
  "manifest_version": 2,
  "description": "PABRE-RW is an editor to write down the
    requirements specifications of a software system
    with the help of a catalogue of SRP. ",
  "author": "GESSI UPC and Núria Bertran",
  "icons": {"128": "smalllogo.png"},
  "app": {
    "background": {
      "persistent": false,
      "scripts": ["background.js", "main.dart.js"]
    }
  }
}
```

*Figura 14: manifest.json de PABRE-RW*

**Background:**

Aquest fitxer conté les dades que configuren l'esdeveniment que obre l'aplicació. Aquest esdeveniment generarà la pàgina/finestra sobre la qual es mostrarà el HTML de l'aplicació. Aquest fitxer ha de contenir les mesures de la finestra i adreçar el fitxer html que s'utilitzarà.

```
chrome.app.runtime.onLaunched.addListener(function() {
  chrome.app.window.create('index.html', {
    'bounds': {
      'width': screen.availWidth,
      'height': screen.availHeight
    }
  });
});
```

*Figura 15: background.json de PABRE-RW*

### HTML, JS i CSS:

De la mateixa manera que una aplicació per navegador necessita HTML , les apps de Chrome també utilitzen el mateix llenguatge de marcatge per poder executar-se. A més a més, utilitzen JavaScript com a llenguatge de programació i CSS per al disseny.

## 3.2 – Escollir software per l'aplicació

Per veure les tecnologies més eficients es va investigar, i es va veure que es podia desenvolupar l'App en les següents tecnologies:

- *JavaScript*: [26] desenvolupar directament en JavaScript per fer Chrome Apps
- *Dart*: [27] es pot compilar per fer Apps Chrome ja que al final es transformarà en JavaScript.
- *CoffeeScript*: [28] Semblant a *Dart*, es pot fer servir en apps Chrome ja que finalment es compilarà en JavaScript.

Es va decidir desenvolupar en Dart. Dart [29] és un llenguatge de codi obert, desenvolupat per Google. Aquest llenguatge no pretén substituir a JavaScript, sinó millorar-lo i ser una tecnologia més moderna.

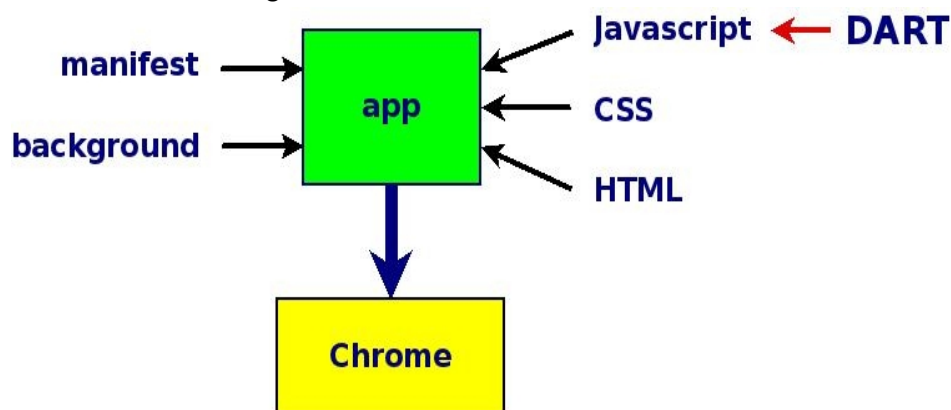


Figura 16: app de Chrome amb Dart

Com es veu a la imatge, ara ja no cal programar directament amb JavaScript, si no que es programa amb Dart que generarà el JavaScript per a l'app.

Per desenvolupar l'aplicació primer es va provar d'utilitzar l'entorn de desenvolupament *Chrome Dev Editor*, però no suportava *Dart*. Finalment es va utilitzar l'entorn *WebStorm*. Aquest és un programa molt còmode per desenvolupar aplicacions web.

Després de baixar *WebStorm* [33], aquest inicialment executava l'aplicació en un navegador normal. Per utilitzar aquest programa, va caldre descarregar una extensió de *Dart* [30]

Primer de tot cal crear un nou projecte de Dart en WebStorm:

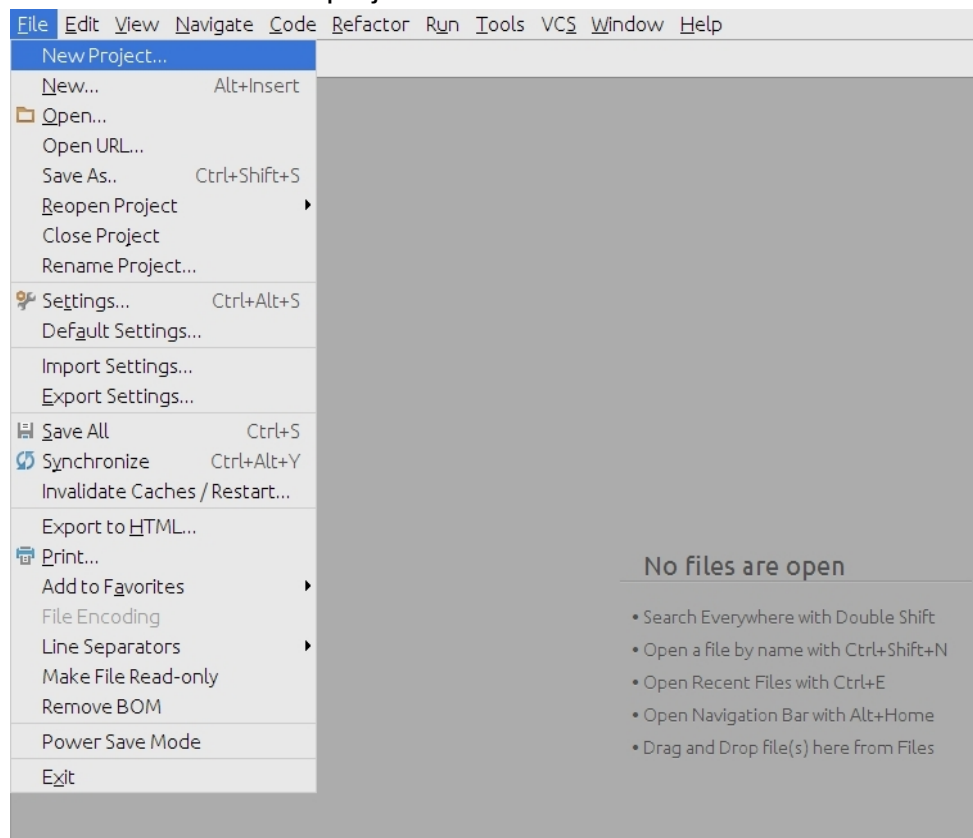


Figura 17: Nou projecte

Un cop escollida l'opció, s'ha d'escollir quin tipus de projecte es vol. En aquest cas es vol que sigui projecte Dart.

S'ha d'escollir la ruta del projecte i quina és la ruta de Dart-sdk, és a dir, l'extensió de Dart que es necessita. En aquesta finestra es pot agafar exemples ja generats. Per acabar es prem el botó "Create" i ja es disposa d'un nou projecte. A la figura es mostra la finestra explicada.

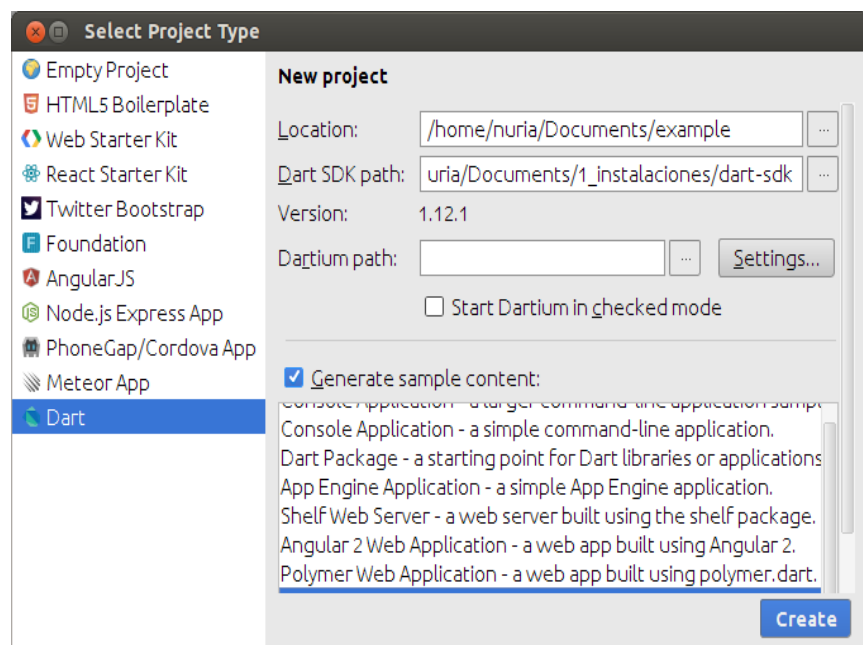


Figura 18: Selecció de tipus de projecte



D'aquesta manera, en el WebStorm ja es disposa de tot el que cal per desenvolupar l'aplicació. Després de funcionar correctament en el navegador normal, s'ha de fer un *Build* del fitxer *pubspec.yaml* perquè l'aplicació tingui els fitxers necessaris perquè s'executi en *Chrome Apps & Extensions Developer Tool* [31]. D'aquesta manera es podrà executar com una App de Chrome a partir del *manifest*.

Per instal·lar el *Chrome Apps & Extensions Developer Tool* cal, des d'un navegador Chrome, accedir a la URL definida a [31] per baixar-lo:

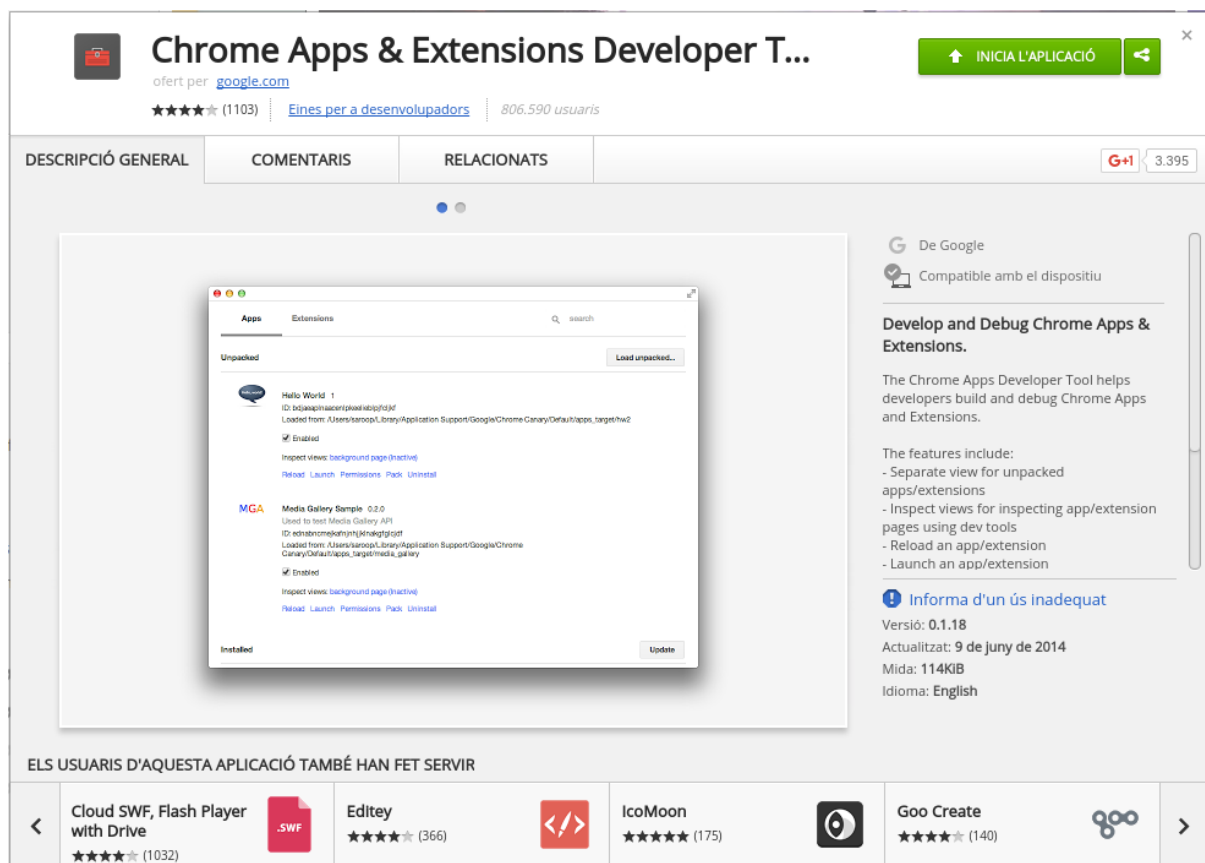


Figura 19: Imatge per baixar l'app "Chrome Apps & Extensions Developer Tool"

A partir de aquí apareixerà a "les meves apps" del navegador Chrome la icona:



Chrome Apps & Exte...

Figura 20: Icona de l'app "Chrome Apps & Extensions Developer Tool"

En aquest punt va sorgir un problema: no s'executava correctament l'aplicació. Tota la part de JavaScript que s'havia transformat no s'executava correctament i donava errors. Va ser un problema del qual va ser difícil trobar la solució perquè no quedava molt clar d'on provenia l'error. Es va pensar que s'havia d'incloure alguna cosa més en el fitxer manifest.json. En el manifest.json es va provar si l'error es produïa perquè no es podia accedir a les pàgines webs del sistema PABRE-WS. Però es va arribar a trobar l'error i tractava sobre la transformació de Dart a JavaScript. Resulta que la transformació de Dart a JavaScript no era total. Es va solucionar canviant una opció en el fitxer que transforma el codi Dart a JavaScript.

En el fitxer *pubspec.yaml* s'havia de posar l'opció `csp: true` : [32]

```
name: project
description: PABRE-RW, app Chrome
transformers:
- $dart2js:
  csp: true
```

Figura 21: *pubspec.yaml*

La solució era que algunes aplicacions, per exemple Chrome apps i apps que funcionen com extensions de Chrome, han de generar el CSP (política de contingut de seguretat) versió de JavaScript. El programador pot fer això assignant el valor *true* al camp *csp* de *dart2js*.

### 3.3 – Organització de l'entorn

En obrir un nou projecte amb el WebStorm es crea una carpeta que contindrà les dades del projecte.

Sol ser convenient i útil agafar algun exemple de Dart per evitar haver de construir-ho tot des de zero. En fer això, al directori del projecte apareixen algunes entrades automàticament. El desenvolupador ha d'acabar de posar l'estructura restant del projecte. L'aspecte final d'aquest directori serà el següent:

Name	Size	Type	Date Modified
▶ build	1 item	folder	Mon Jan 11 20:17:07 2016
▶ packages	36 items	folder	Mon Jan 11 20:16:56 2016
▶ web	9 items	folder	Mon Jan 11 20:16:56 2016
pubspec.lock	2.9 kB	plain text document	Mon Jan 11 20:16:56 2016
pubspec.yaml	140 bytes	YAML document	Mon Nov 2 00:54:22 2015

Figura 22: Carpets i fitxers dintre del directori del projecte

A la carpeta *web* hi ha tots els fitxers per desenvolupar l'aplicació en un navegador normal.

Els fitxers originals del desenvolupador són al directori *web*. Un cop traduït a Javascript, la traducció va a parar al directori *build*, en el qual s'hi copia de forma automàtica tota la carpeta *web* a més de posar-hi els fitxers per executar l'app com una app de Chrome. Per fer la traducció i crear el directori *build* cal fer un *Build* del fitxer *pubspec.yaml*.

Name	Size	Type	Date Modified
▶ build	1 item	folder	Mon Jan 11 20:17:07 2016
▶ packages	36 items	folder	Mon Jan 11 20:16:56 2016
▼ web	9 items	folder	Mon Jan 11 20:16:56 2016
▶ Templates	4 items	folder	Mon Jan 11 20:16:56 2016
▶ botons	18 items	folder	Mon Jan 11 20:16:56 2016
▶ packages	36 items	Link to folder	Mon Jan 11 20:16:56 2016
background.js	171 bytes	JavaScript program	Sat Dec 26 11:07:41 2015
index.html	4.6 kB	HTML document	Mon Jan 11 20:16:33 2016
main.dart	24.9 kB	plain text document	Mon Jan 11 17:15:32 2016
manifest.json	270 bytes	plain text document	Mon Dec 21 08:47:34 2015
smalllogo.png	8.2 kB	PNG image	Tue Nov 3 11:31:44 2015
styles.css	1.9 kB	CSS stylesheet	Mon Jan 11 20:06:29 2016
pubspec.lock	2.9 kB	plain text document	Mon Jan 11 20:16:56 2016
pubspec.yaml	140 bytes	YAML document	Mon Nov 2 00:54:22 2015

Figura 23: Carpetes i fitxers dintre del directori *web*

A la següent figura es mostra el contingut de la carpeta *build*:

Name	Size	Type	Date Modified
▼ build	1 item	folder	Mon Jan 11 20:17:07 2016
▼ web	9 items	folder	Mon Jan 11 20:17:07 2016
▶ Templates	3 items	folder	Mon Jan 11 20:17:07 2016
▶ botons	17 items	folder	Mon Jan 11 20:17:07 2016
▶ packages	4 items	folder	Mon Jan 11 20:17:07 2016
background.js	171 bytes	JavaScript program	Mon Jan 11 20:17:07 2016
index.html	4.6 kB	HTML document	Mon Jan 11 20:17:07 2016
main.dart.js	233.9 kB	JavaScript program	Mon Jan 11 20:17:07 2016
manifest.json	270 bytes	plain text document	Mon Jan 11 20:17:07 2016
smalllogo.png	8.2 kB	PNG image	Mon Jan 11 20:17:07 2016
styles.css	1.9 kB	CSS stylesheet	Mon Jan 11 20:17:07 2016
▶ packages	36 items	folder	Mon Jan 11 20:16:56 2016
▶ web	9 items	folder	Mon Jan 11 20:16:56 2016
pubspec.lock	2.9 kB	plain text document	Mon Jan 11 20:16:56 2016
pubspec.yaml	140 bytes	YAML document	Mon Nov 2 00:54:22 2015

Figura 24: Carpetes i fitxers dintre del directori *build*

La carpeta *packages* conté automàticament totes les extensions necessàries per executar el codi del programa.

## 4 – Primera versió de PABRE-RW

En aquest capítol s'explica com s'ha assolit la tercera tasca corresponent a l'objectiu 3 (explicat en el capítol 1 secció “**Abast del projecte**”).

Desenvolupar una primera versió de PABRE-RW que permeti escriure requisits, guardar aquests requisits en una base de dades local.

### 4.1 – Interfície de l'aplicació

A l'inici del desenvolupament de l'aplicació, es va dissenyar la interfície. Principalment hi havia dues parts:

- Àrea d'edició: Aquí aniran tots els camps que es necessiten per poder escriure text, guardar en un fitxer el text i pujar un nou fitxer. A més a més, de poder editar el text.
- Àrea de recerca: Aquí aniran totes les dades que s'agafen de les dades del sistema PABRE (PABRE-WS).



*Figura 25: Disseny bàsic de la interfície*

Un cop s'engega l'app, en la interfície apareix el següent disseny:

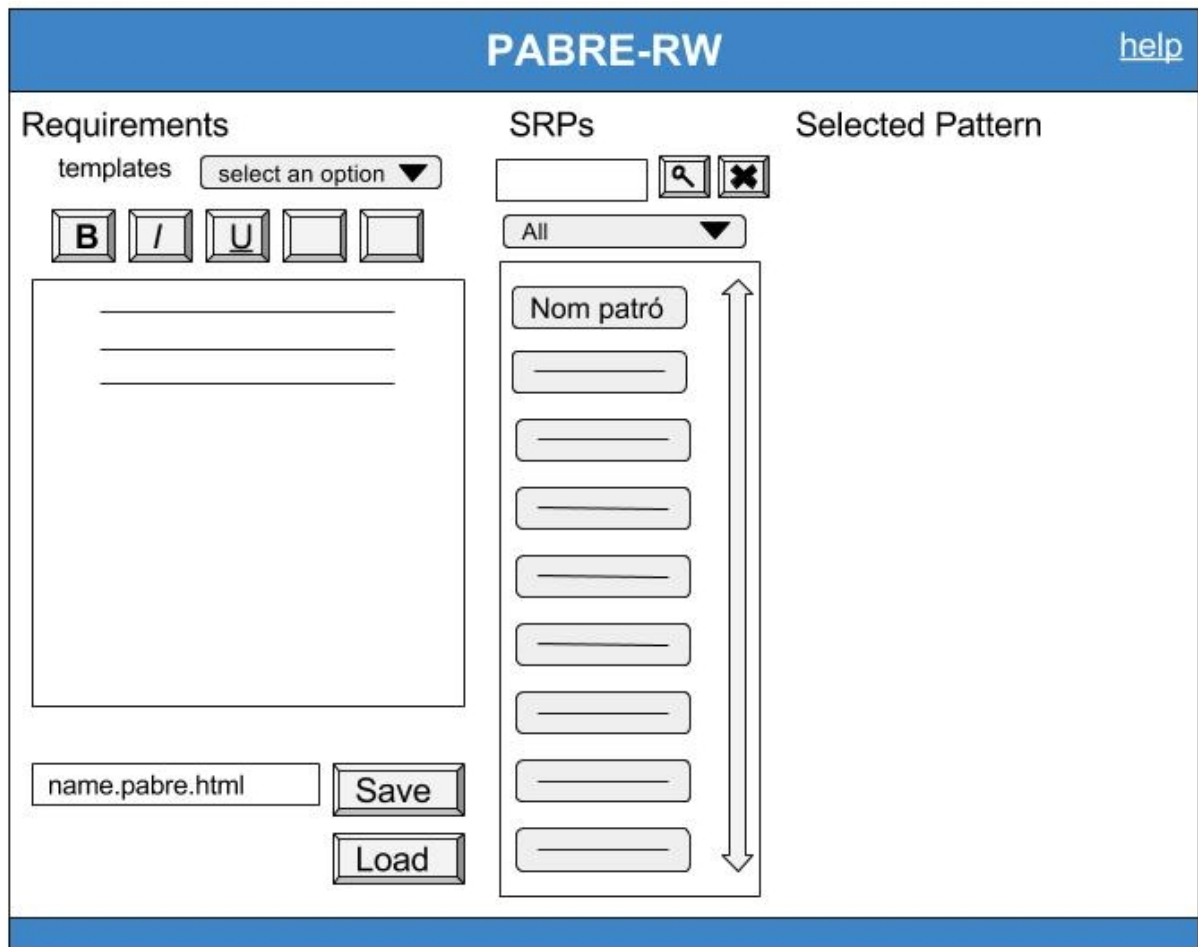


Figura 26: Disseny quan s'engega la interfície

En aquesta figura es poden veure les següents columnes:

- **Requirements** (àrea d'edició): Aquí hi ha els projectes ja creats (templates). També hi ha l'editor de text on l'usuari podrà escriure tots els requisits que vulgui i els botons per canviar el disseny d'aquest text. A més a més, a baix de l'editor hi ha dos botons per guardar el contingut a un fitxer i pujar un fitxer nou a l'aplicació.
- **SRPs** (àrea de recerca): Aquí hi ha la llista dels noms dels patrons de requisits que seran visibles en l'aplicació. Cada nom és un botó que mostra el contingut d'aquell patró de requisits, en el apartat *Nom Patró*. A sobre de la llista de patrons hi apareixen dos mètodes de recerca de patrons, que poden escurçar la llista mostrada: cerca per Classifier Schemas (el desplegable que a la figura mostra "All"), i el caxetí amb dos botons al costat (lupa i creu) per fer filtratge per paraules.
- **Selected Pattern** (àrea de recerca): aquí apareixerà la informació que tingui el patró seleccionat.

Un cop es seleccioni un botó **Nom patró** de la columna SRPs, apareixerà la informació del patró en la columna Selected Pattern, com es veu en la següent figura:

The screenshot shows the PABRE-RW web application interface. It is divided into three main sections: Requirements, SRPs, and Selected Pattern.

- Requirements:** Contains a 'templates' section with a 'select an option' dropdown and five buttons labeled B, /, U, and two empty boxes. Below this is a large text area with three horizontal lines. At the bottom, there is a text input field containing 'name.pabre.html' and two buttons labeled 'Save' and 'Load'.
- SRPs:** Contains a search bar with a magnifying glass icon and a close icon. Below it is a dropdown menu set to 'All'. A list of buttons follows, with the first one labeled 'Nom patró' and the others being empty. A vertical double-headed arrow is positioned to the right of this list.
- Selected Pattern:** Contains a section titled 'Nom patró' with a 'Descripció Objectiu' label and a text input field containing 'Nom form - descripció'. A vertical double-headed arrow is positioned to the right of this input field.

*Figura 27: Disseny de la interfície seleccionant un patró*

Com es pot veure, el patró conté el nom, objectiu, la descripció i els diferents formularis que tingui i les seves descripcions.

Un cop es seleccioni el botó **Nom form - descripció**, apareixerà la informació del form en la columna Selected Pattern.

Això es pot veure en la figura de la pàgina següent:

The screenshot shows the PABRE-RW web application interface. The top bar is blue with the title 'PABRE-RW' and a 'help' link. The main content area is divided into three columns: 'Requirements', 'SRPs', and 'Selected Pattern'.

- Requirements:** Contains a 'templates' section with a 'select an option' dropdown and five buttons labeled 'B', 'I', 'U', and two empty boxes. Below this is a large text area with three horizontal lines. At the bottom, there is a text input field containing 'name.pabre.html' and two buttons: 'Save' and 'Load'.
- SRPs:** Contains a search bar with a magnifying glass icon and a close icon. Below it is a dropdown menu set to 'All'. A list of 'Nom patró' items is shown, each with a text input field. A vertical double-headed arrow indicates scrolling.
- Selected Pattern:** Contains a section for 'Nom patró' with a 'Descripció Objectiu' label and a text input field. Below this is a section for 'Nom form' with a 'Nom part - descripció' label and a text input field. A vertical double-headed arrow indicates scrolling.

Figura 28: Disseny de la interfície seleccionant un form

Com es pot veure, el form conté el nom, les diferents parts que tingui i les seves descripcions.

L'estructura mostrada fa el disseny compacte i agrupa en una columna totes les dades d'un patró de manera coherent. Però es fa a costa de forçar el scrolling quan un pattern contingui molts forms o un form contingui moltes parts. Aquest scrolling pot ser una mica incòmode en no deixar veure d'un cop d'ull tot el contingut, però s'ha preferit fer-ho així per la unitat conceptual que proporciona a la columna i a tota l'app. L'alternativa seria abandonar la finestra única i obrir una finestra diferent per a cada columna actual, cosa que té els seus propis inconvenients i afectaria l'intenció bàsica de mantenir aquesta app el més senzilla possible

Un cop es seleccioni el botó  apareixerà la informació de les variables que conté aquella part i quedarà de la següent manera:



Figura 29: Disseny de la interfície seleccionant una part

## 4.2 – Escriure Requisites

Aquesta subtasca fa referència al cas d'ús **“Editar requisits”**. Com bé s'ha comentat en el cas d'ús, es vol escriure requisits en un editor de text. De moment, en aquesta versió, només s'afegeix, modifica i elimina informació de l'editor de text.

Per escriure requisits es va crear un Textarea per posar el text. El Textarea és un element de HTML. Aquest element és una entrada de text de diverses línies. Aquesta àrea pot contenir una quantitat il·limitada de caràcters. A més, té una amplada fixa i es pot canviar el disseny amb CSS.

## 4.3 – Guardar requisits

Aquesta subtasca fa referència al cas d'ús **“Guardar requisits”**. Com bé s'ha comentat en el cas d'ús, es vol guardar un projecte en local.

En aquest objectiu es va tenir la primera idea de guardar els requisits en la BD de l'API de Chrome. Finalment es va decidir guardar-lo en un fitxer local en l'ordinador.



Així el programa gestionaria les dades i no faria falta guardar-les en la BD de l'API de Chrome

Per guardar requisits s'ha tingut el següent problema: no s'ha pogut decidir la ruta que vulgui l'usuari. És a dir, els fitxers es guarden localment en la carpeta que té l'usuari per defecte en el navegador de Chrome, normalment en baixades. Una possible solució era utilitzar el mètode `exeCommand`. Aquest mètode permet executar ordres per a una regió editable. El problema és que el mètode "Save As" no està disponible per a Chrome.

També es va tenir un segon problema. Quan es guarda per defecte, l'usuari no sap si s'ha guardat o no el fitxer. És per això que s'ha utilitzat el mètode `window.alert()`. Aquest mètode crea un pop-up emergent per a que l'usuari sàpiga que s'ha baixat el fitxer. El problema va ser quan no funcionava com a App de Chrome. En l'App de Chrome el mètode `window.alert()` està inhabilitat. Per solucionar aquest problema es van crear diversos elements `dialog` de HTML. A més a més vaig crear un altre dialog en el cas de no guardar-se el fitxer amb l'extensió correcta, és a dir, .pabre o .txt.

## 5 – Segona versió de PABRE-RW

En aquest capítol s'explica com s'ha assolit la quarta tasca corresponent a l'objectiu 4 (explicat en el capítol 1 secció “**Abast del projecte**”).

Aquesta segona versió millora l'anterior. Es vol millorar la part d'editar requisits i guardar-los altre cop modificats. A més a més s'afegeix la part de recuperar des de la base de dades local un fitxer; aquest fitxer conté requisits que es mostraran a l'àrea de text de l'aplicació.

### 5.1 – Editar els requisits

Aquesta subtasca fa referència al cas d'ús “**Editar requisits**”. Com bé s'ha comentat en el cas d'ús, es pretén escriure requisits en un editor de text. En la versió anterior només es tenia en compte afegir, modificar i eliminar informació de l'editor de text. En aquesta versió, s'afegeixen botons per poder posar un editor amb opcions per canviar el disseny de l'editor. Per a que es pugui escollir una opció, sempre s'ha d'haver seleccionat el text anteriorment per saber de quina part s'ha de canviar el tipus d'opció.

S'han creat les següents opcions per modificar el disseny de l'editor:

- Negreta: la informació es pot posar en negreta
- Cursiva: la informació es pot posar en cursiva
- Subratllat: la informació es pot posar en subratllat
- Ratllat: la informació es pot posar en ratllat
- Llista de punts: la informació es pot posar en una llista de punts
- Llista de números: la informació es pot posar en una llista de números
- Indentació: la informació es pot indentar
- Desindentar: la informació es pot desindentar
- Enrere: permet recuperar la informació prèvia a la darrera acció realitzada.
- Endavant: permet recuperar la informació prèvia a la darrera acció realitzada si aquesta ha estat un Enrere.

Al disseny de l'aplicació s'ha tingut en compte que aquests botons fossin estàndards per tal d'ajudar a l'usuari. Es mostren a la següent figura:



Figura 30: Icones de l'editor de text

En passar a text enriquit es planteja el format en el que treballar i desar aquest text enriquit. L'elecció natural, donat que estem en un entorn HTML, és usar un sotsconjunt dels tags HTML per indicar cada una de les característiques que es poden implementar amb els botons abans esmentats.

Per editar els requisits en aquesta versió s'ha canviat el *Textarea* per un *div* amb `ContentEditable=True`. *Div* és un element de HTML que s'utilitza per fer una divisió o secció en el document. Aquest element no és un text editable com *Textarea*, per això s'ha de posar l'opció `ContentEditable=True` perquè, a l'igual que el *Textarea*, s'hi puguin editar caràcters. Es va haver d'afegir algunes altres opcions amb el CSS, ja que no estaven per defecte. Per exemple s'havia de posar scroll automàtic, quan en el *Textarea* ja estava per defecte. Es va canviar l'element *Textarea* per l'element *Div* ja que el *Textarea* no permetia posar un editor de text ric, és a dir, amb negreta, cursiva, subratllat, etc.

D'aquesta manera es pot treballar amb un text enriquit a l'estil dels tractaments de text amb les opcions bàsiques. Per això es va escollir el mètode `execCommand`. Com s'ha comentat anteriorment, aquest mètode permet executar ordres per a una regió editable. Té la següent sintaxi:

```
bool=document.execCommand(Command,Default,Argument)
```

- `bool`: el booleà et diu si està suportat i si està activat o no.
- `Command`: el nom de l'ordre que es vol executar. Les comandes que utilitzo són les següents:
  - `bold` per la negreta.
  - `italic` per la cursiva.
  - `underline` per al subratllat.
  - `strikeThrough` per al ratllat.
  - `insertunorderedlist` per a la llista de punts.
  - `insertorderedlist` per a la llista de números.
  - `indent` per a la indentació.
  - `outdent` per a la desindentació.
  - `undo` per desfer.
  - `redo` per tornar a fer.
- `Default`: un booleà que diu si per defecte s'ha de mostrar. Li passo `true`.
- `Argument`: si es necessita, el valor de l'argument d'entrada. Li passo una cadena buida.

## 5.2 – Guardar els requisits

Aquesta subtasca fa referència al cas d'ús “**Guardar requisits**”. Com bé s'ha comentat en el cas d'ús, es vol guardar els requisits en la BD local.

Per guardar els fitxers a la base de dades local no es va canviar el mètode de guardar però sí la seva extensió. Això va ser perquè al editar el text es va canviar de format: de text pla es va passar a HTML per a que fos un text ric. De manera que es va canviar el format del fitxer `.txt` o `.pabre` a `.pabre.html` per dues raons:

- Es posa l'extensió *.html* perquè per editar el text es creen tags de HTML. S'aprofita pat de la sintaxi HTML per codificar el text enriquit. A més, aquesta extensió permet que el Dart reconegui els tags HTML que contindrà el fitxer
- Cal posar *.pabre* per assegurar que són fitxers creats per l'aplicació i per tant que contindran només els tags HTML que l'aplicació pugui suportar.

L'aplicació controla que l'extensió sigui la correcta.

### 5.3 – Recuperar els requisits

Aquesta subtasca fa referència al cas d'ús “**Recuperar requisits**”. En aquesta versió s'afegeix aquest cas d'ús, tracta d'agafar un fitxer del disc de l'ordinador i mostrar-lo en l'editor de text per poder modificar-lo.

Per recuperar els requisits des d'una base de dades local, s'ha afegit un botó que obre una finestra i permet navegar per les carpetes de l'ordinador, és a dir per la base de dades local.

El problema va ser poder agafar el contingut del fitxer, ja que un cop pujat el fitxer només es tenia el nom i tamany del fitxer, però no el contingut. Es va intentar agafar la ruta de l'ordinador però també desapareixia. Cal tractar el fitxer mentre s'està pujant, un cop s'ha llegit tot ja no es pot accedir al contingut.

Finalment es va cridar al mètode *onLoad*. Aquest mètode crida a una funció quan encara s'està pujant un fitxer. Quan es cridava aquest mètode s'executava una funció, mentre s'estava executant el mètode el fitxer encara no s'havia pujat el tot i això va permetre accedir-ne al contingut.

L'extensió de pujar fitxers ha de ser la mateixa que la de guardar, és a dir, *.pabre.html*. A l'igual que en la secció 5.2, també es controla l'error que l'extensió no sigui la correcta.

### 5.4 – Interfície

Per als botons de guardar i pujar fitxer també s'han usat els botons estàndards, tal com s'ha comentat en la secció 5.1 – *Editar requisits*.

En aquesta versió, la interfície va quedar de la següent manera:

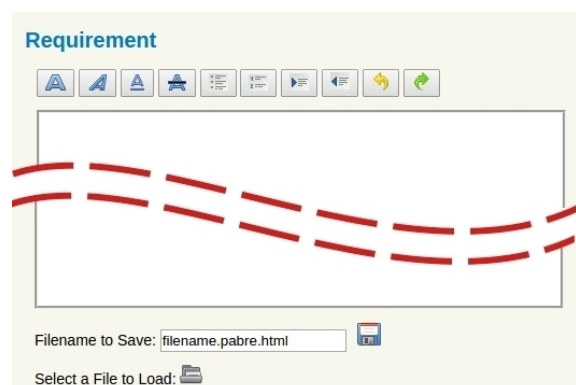


Figura 31: Interfície de l'editor de text

## 6 – Tercera versió de PABRE-RW

En aquest capítol s'explica com s'ha assolit la cinquena tasca corresponent a l'objectiu 5 (explicat en el capítol 1 secció “**Abast del projecte**”).

A la tercera versió de PABRE-RW s'afegeix una nova àrea en la que es mostren els SRPs i el contingut de cada un d'ells. Aquesta versió fa referència al cas d'ús “**Accedir als SRPs**”.

Aquesta versió afegeix una nova àrea a la interfície on mostra tots els SRPs i la seva informació.

### 6.1 – Afegir els SRPs

Primer de tot es necessita accedir a la base dades PABRE (PABRE-WS). Al capítol 2 s'ha explicat com s'hi pot accedir. Un cop s'ha accedit als serveis web, es mostra una llista dels noms dels patrons que conté PABRE-WS. Cada nom és un botó per poder accedir als seus forms.

### 6.2 – Afegir els Forms dels SRPs

Per afegir els forms que tingui cada un dels patrons, s'ha de prémer el botó del nom del patró. Un cop seleccionat es mostrarà a la part dreta les següents dades:

- El nom del patró
- La descripció del patró
- La descripció del Goal
- Els noms dels forms i les seves descripcions.

Cada form és un botó per poder accedir a les seves parts.

### 6.3 – Afegir les parts dels SRPS

Per afegir les parts que té cada form de un patró, s'ha de prémer el botó del nom o descripció del form. Un cop seleccionat es mostrarà a sota el següent:

- El nom del form
- La descripció del form
- La descripció de la part fixa
- El nom i descripció de la part estesa

### 6.4 – Usabilitat

Per temes d'usabilitat es van fer alguns canvis. Inicialment cada nom de la llista tenia forma de botó. Això també va passar amb els forms i les parts. Es va canviar i es va fer una llista sense forma de botons.

Com que l'aparença de botons feia molt farragosa aquesta llista, es va decidir mostrar una llista de paraules on cada paraula és en el fons un botó des del punt de vista del programa, però l'usuari només veu els noms dels SRPs.

Quan es passa per sobre del nom del SRP es marca que es pot seleccionar, tot subratllant-lo. A més, un cop es prem un d'aquest botons camuflats, aquest es queda marcat amb un color diferent per a que l'usuari se n'adoni. Tot això es va fer igual per als form i per a les parts.

En la següent figura es pot veure el disseny del botó inicialment (esquerra) i com ha quedat finalment (dreta) la llista dels Patterns Catalogue:

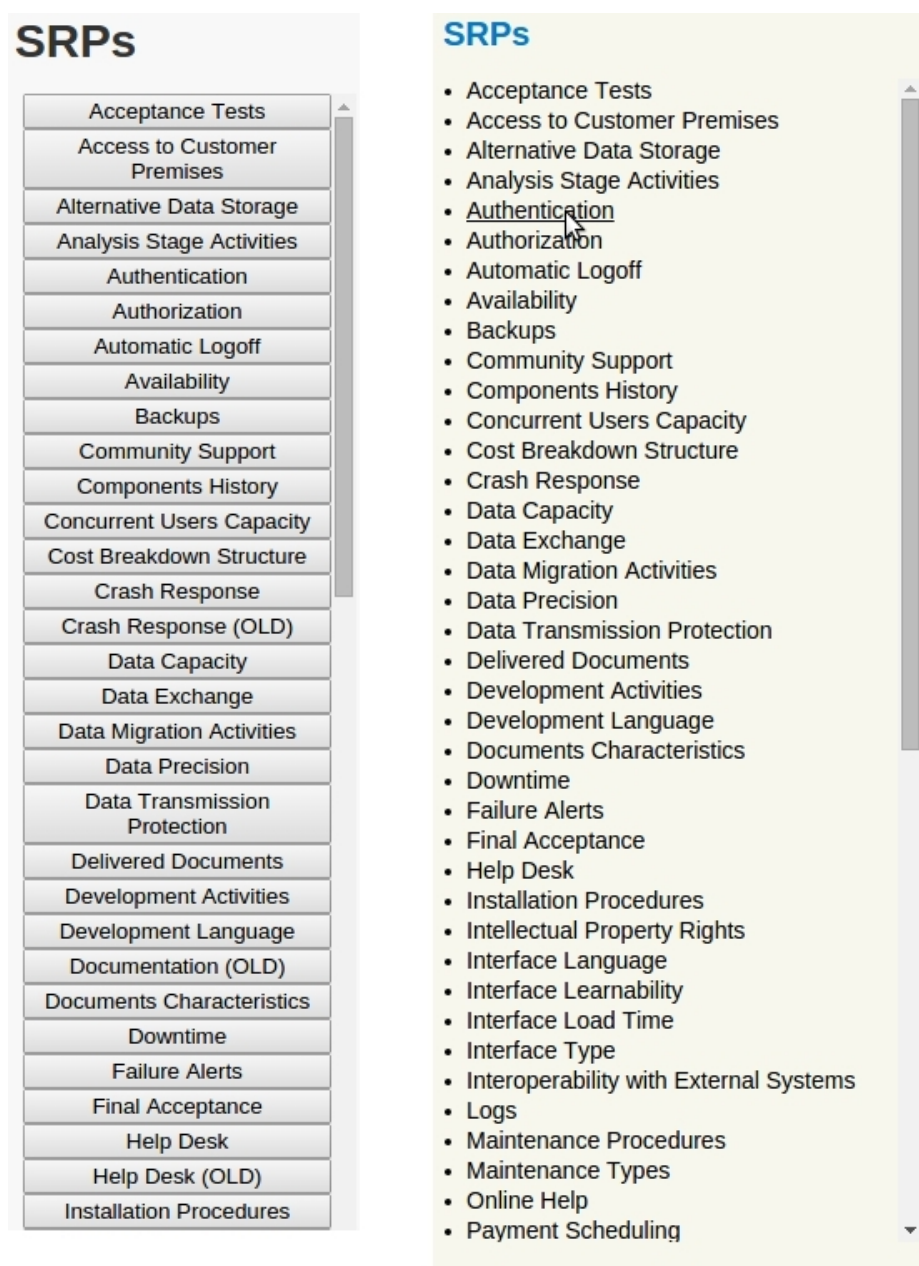


Figura 32: Comparació dels botons dels noms dels SRPs

Finalment, aquest cas d'ús es mostra amb les següents figures. Inicialment, tal com s'indica a la dreta (Advice), cal seleccionar un SRP de la llista del mig:

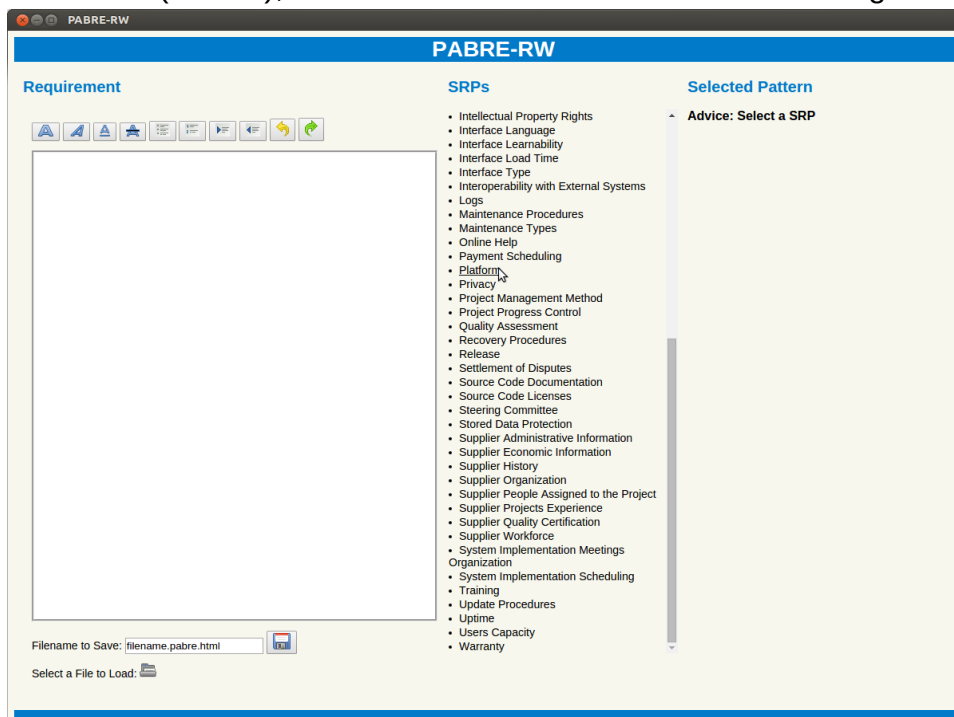


Figura 33: Llista dels SRPs

Un cop triat el SRP apareixen els seus forms a la dreta: com diu l'advice, en podem triar un:

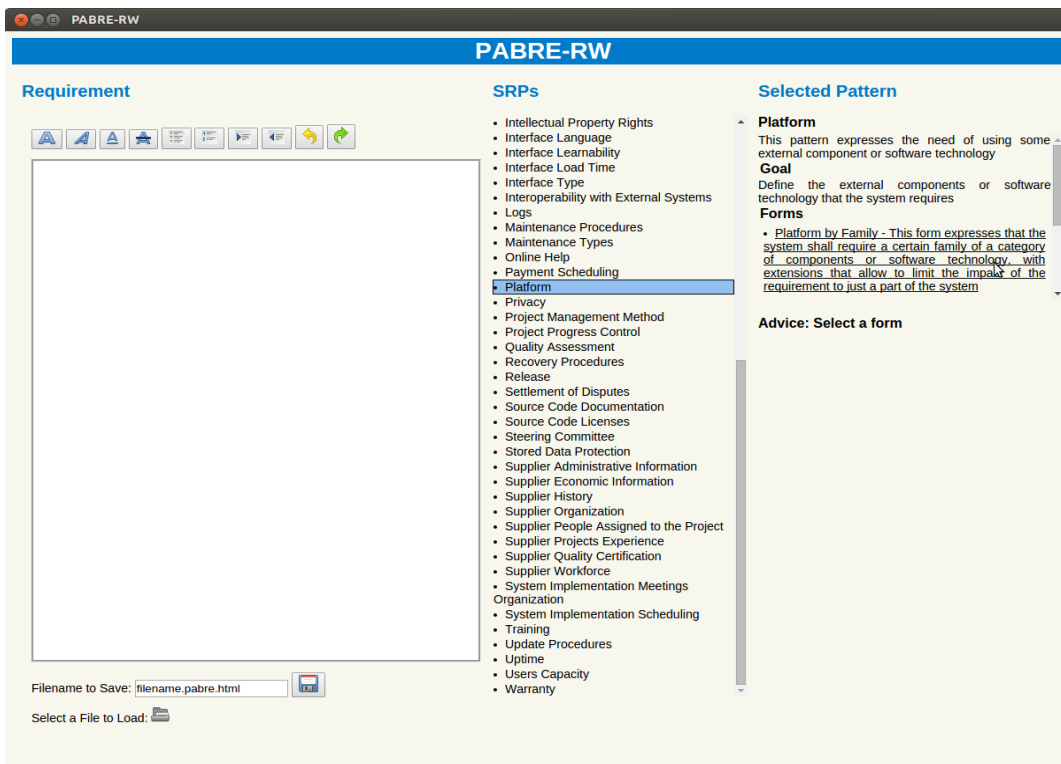


Figura 34: Selecció d'un SRP i mostrar els forms

En clicar un form es mostren a sota les parts tant fixes com esteses:

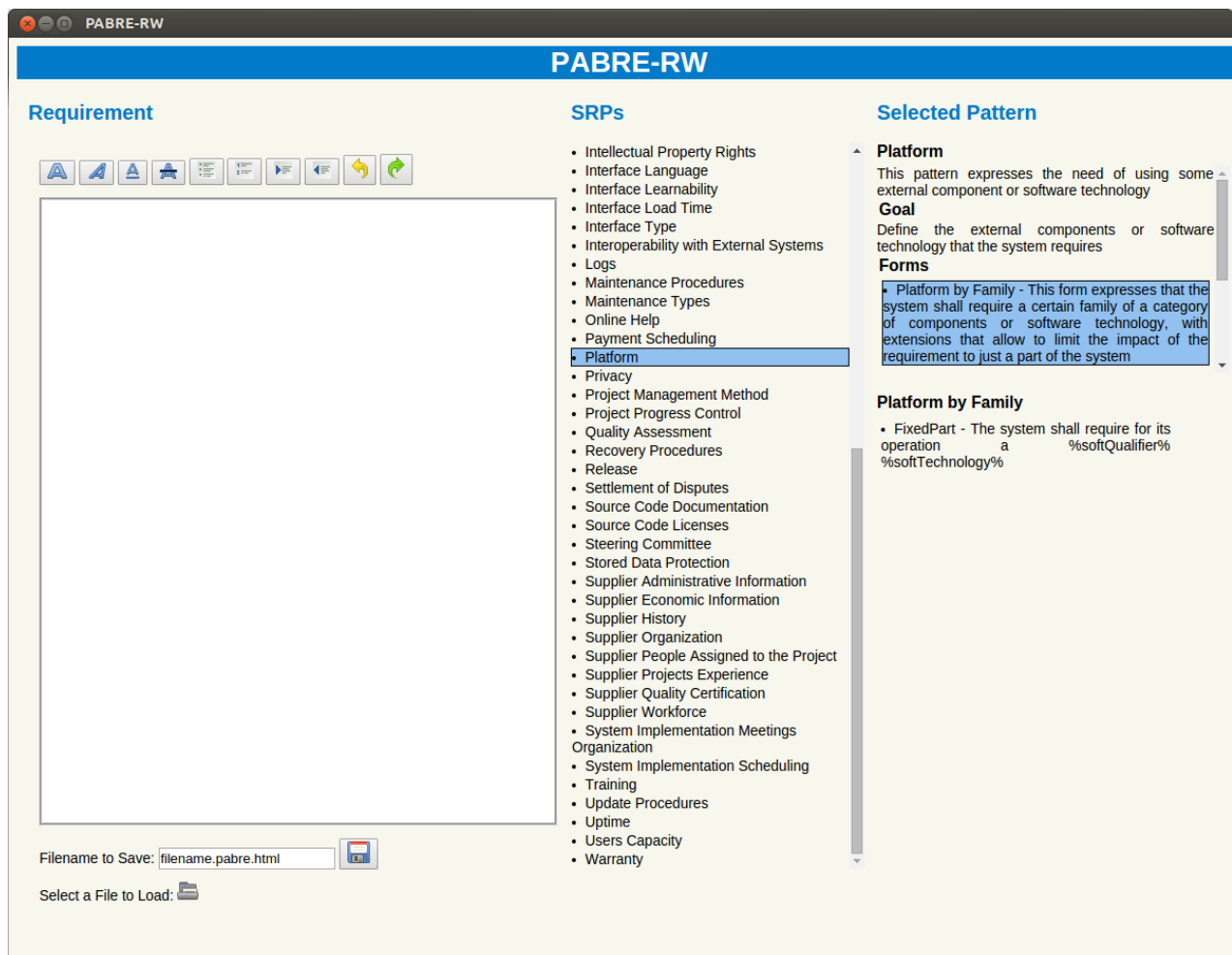


Figura 35: Selecció d'un form i mostrar les parts



## 7 – Quarta versió de PABRE-RW

En aquest capítol s'explica com s'ha assolit la sisena tasca corresponent l'objectiu 6 (explicat en el capítol 1 secció “**Abast del projecte**”).

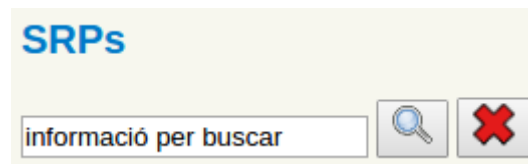
La quarta versió de PABRE-RW afegeix facilitats de cerca/ordenació dels SRPs i el contingut de cada un d'ells

### 7.1 – Cerca dels SRPs

Es vol facilitar la feina a l'enginyer de requisits a l'hora de trobar el SRP que vol aplicar en un projecte. Això es farà oferint funcionalitats de cerca. La cerca d'un SRP es farà tenint en compte les paraules que hi ha al goal, keywords i descripció del patró, dels form i les parts d'un SRP.

Aquesta funcionalitat fa referència al cas d'ús “**Cercar dels SRPs**”. Com bé es pot veure, en aquesta funcionalitat primer de tot s'ha obtingut la informació que l'usuari pot cercar. Un cop es té la informació, l'usuari cerca informació i li diu al sistema que mostri en la llista només els noms dels SRPs que contenen aquesta informació.

De la mateixa manera que els botons emprats durant l'edició de requisits (veure secció 5.1 - *Editar requisits*), també s'han afegit els botons estàndards per la cerca dels SRPs:



*Figura 36: Cerca de paraules*

Com es pot veure en la figura hi ha tres camps. El primer és per escriure la informació. El segon és per cercar aquesta informació, i el tercer per netejar la possible cerca anterior, per tal que així torni a mostrar els noms de tots els SRPs.

## 8 – Cinquena versió de PABRE-RW

En aquest capítol s'explica com s'ha assolit la setena tasca corresponent l'objectiu 7 (explicat en el capítol 1 secció “Abast del projecte”).

En la cinquena versió es vol seleccionar una part i usar-la en l'editor.

### 8.1 – Selecció d'una part d'un SRP

Aquesta funcionalitat fa referència al cas d'ús “Insertar una part a l'editor”. Ha de permetre que l'usuari seleccioni una part d'un patró (la part és una frase) i que aquesta part s'afegeixi al projecte com a requisit. Per afegir aquesta part, es té en compte que hi ha variables, que es posaran en negreta per facilitar de l'usuari. A més, es posarà informació per ajudar a l'usuari a esbrinar quin contingut poden tenir aquestes variables.

En aquest punt es va canviar la interfície, ja que es va veure que la informació de les variables quedava molt lluny de l'editor. Per tant es va canviar per facilitar-ne l'ús.

Les següents figures mostren la nova interfície. Partim d'un form triat i veiem un *advice* que suggereix seleccionar una part:

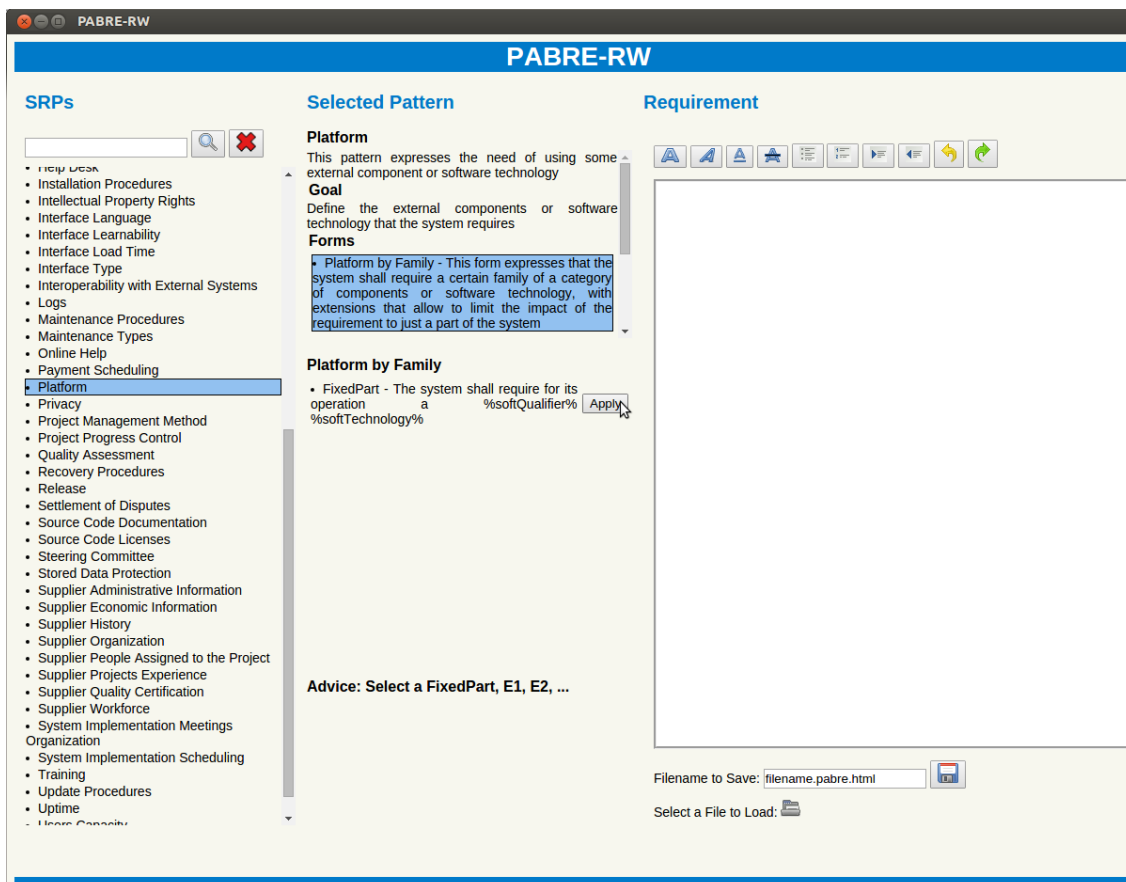


Figura 37: Llista de les parts

Cal observar que es pot clicar la part directament o bé prémer el botó que hi ha a la seva dreta.

En ambdós cassos apareix la informació del quadre *Variable/Possible value* a la part inferior de l'àrea *Selected Pattern*.

En el cas que s'hagi clicat el botó *Apply*, a més del quadre inferior, el requisit apareix a l'àrea d'edició (*Requirement*)

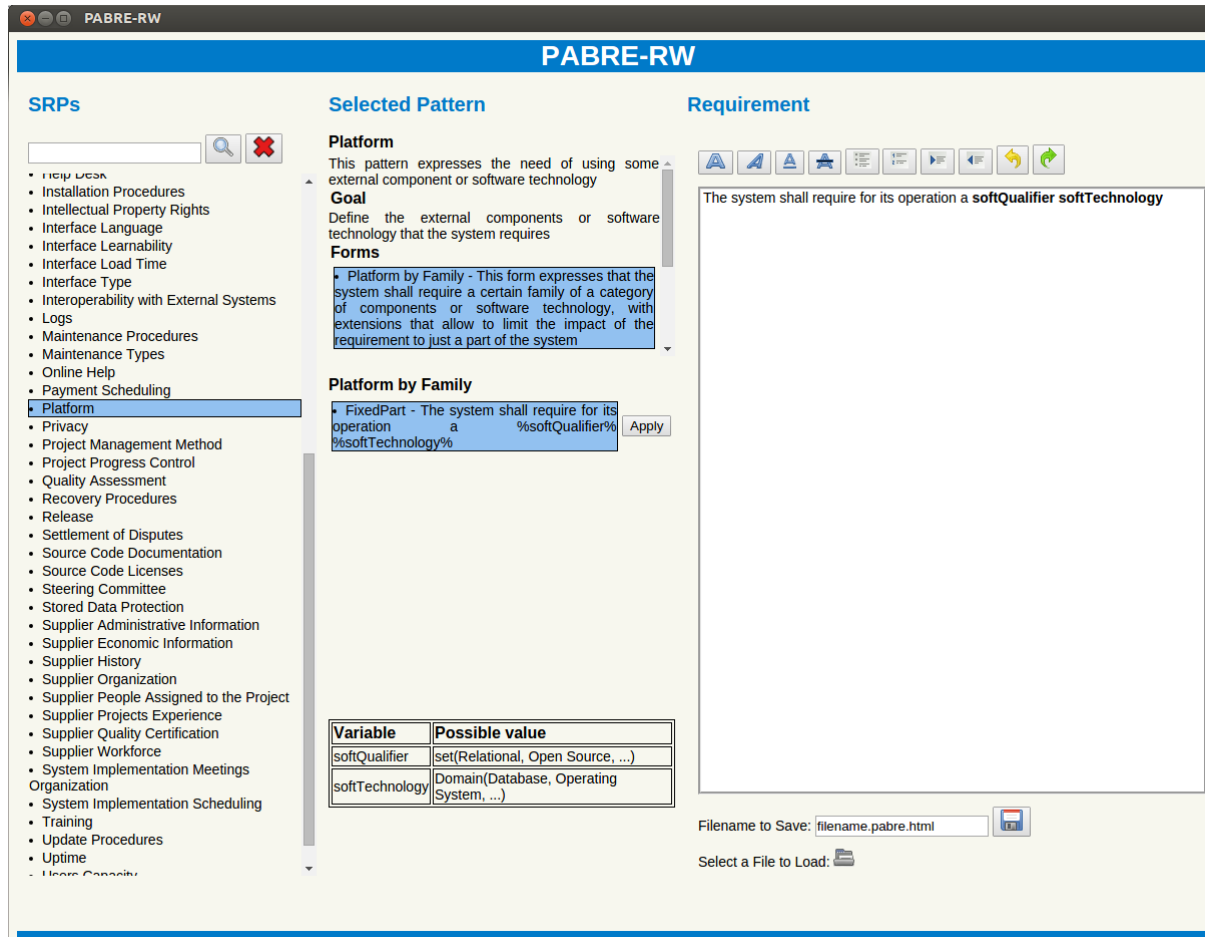


Figura 38: Selecció d'una part

## 9 – Sisena versió de PABRE-RW

En aquest capítol s'explica com s'ha assolit la vuitena tasca corresponent l'objectiu 8 (explicat en el capítol 1 secció “Abast del projecte”).

En una sisena versió de PABRE-RW hi ha un esquema i es pot seleccionar una classificació per veure els patrons que conté. A més, s'ha decidit canviar el nom de les columnes per: Patterns Catalogue (antic SRPs), Selected Patterns (es manté igual), Requirements Specification (antic Requirements)

### 9.1 – Cercar un esquema

Semblant al cas d'ús “Cercar dels SRPs”, aquesta versió fa referència al cas d'ús “Seleccionar esquema”. En aquest cas d'ús també es vol facilitar la feina a l'enginyer de requisits a l'hora de trobar el SRP que vol aplicar en un projecte. Això es farà oferint funcionalitats de cerca. En aquesta cerca l'usuari selecciona un classificador d'un esquema, i el sistema mostra els SRPs que conté el classificador.

Per fer-ho s'ha hagut de accedir als esquemes classificadors de la base de dades PABRE-WS. El com estan estructurats i com accedir-hi estan explicats al capítol 2.

Per fer aquesta funcionalitat, es va reduir de la següent manera: la selecció només es pot escollir d'un únic esquema, “ISO /IEC 25010 (Extended NT, NF)”. A més, només s'agafen els classificadors del primer nivell, és a dir, els *rootClassifier*. Si aquell classificador està subdividit en més d'un sots-classificador, no es mostra el sots-classificador. Un cop seleccionat el classificador (*rootClassifier*), es mostren els noms dels patrons que referència.

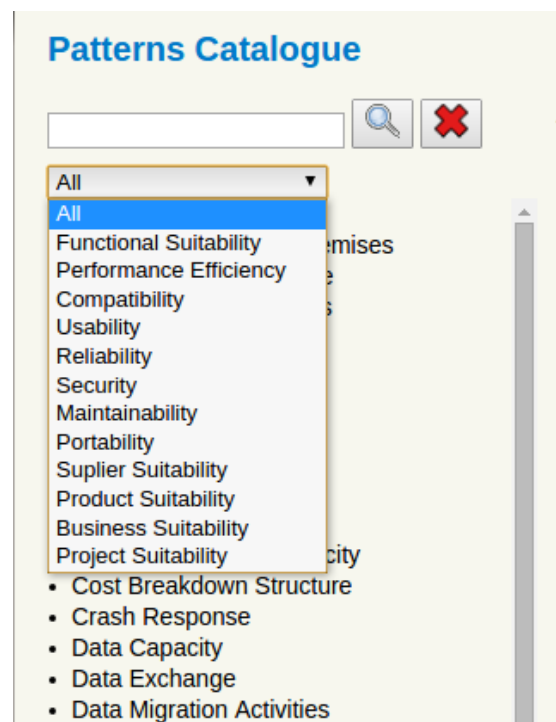


Figura 39: Selecció d'esquema

## 10 – Setena versió de PABRE-RW

En aquest capítol s'explica com s'ha assolit la novena tasca corresponent l'objectiu 9 (explicat en el capítol 1 secció “Abast del projecte”).

La setena versió de PABRE-RW permet seleccionar uns projectes predeterminats, és a dir, prefabricats perquè puguin servir com a exemple (encara que es poden modificar per usar-los). D'aquest projectes en direm *templates* a l'aplicació.

### 10.1 – Selecció d'un projecte

Aquesta funcionalitat fa referència la cas d'ús “Obrir templates”. Tracta sobre un desplegable o l'usuari pot agafar diversos fitxers que ja contenen requisits en el format *pabre.html* usat per l'aplicació. D'aquesta manera li és fàcil a l'usuari utilitzar l'aplicació ja que té unes plantilles per practicar, entendre i usar.

Actualment en el desplegable hi ha els següent fitxers que estan en format *.pabre.html*:

- Blank Sheet: és un fitxer en blanc per netejar la informació.
- Iso25010schema: és un fitxer on es mostren els diferents tipus de requisits que hi ha:
- PABRE-MAN-RequirementsExample: és un fitxer que mostra la mateixa estructura que l'anterior però amb exemples dels requisits.

A part d'aquests fitxers, se'n poden afegir més des del servidor en les noves versions de l'aplicació.

#### Iso25010schema:

REQUIREMENTS SPECIFICATION DOCUMENT

PROJECT CONTEXT AND SCOPE

FUNCTIONAL REQUIREMENTS

1.- Functional Suitability

NON-FUNCTIONAL REQUIREMENTS

2.- Performance Efficiency

3.- Compatibility

4.- Usability

5.- Reliability

6.- Security

7.- Maintainability

8.- Portability

## NON-TECHNICAL REQUIREMENTS

## 9.- Supplier Suitability

## 10.- Product Suitability

## 11.- Business Suitability

## 12.- Project Suitability

Figura 40: Iso25010schema

## PABRE-MAN-RequirementsExample:

## REQUIREMENTS SPECIFICATION DOCUMENT

## PROJECT CONTEXT AND SCOPE

The **reuse of software requirements** may help requirement engineers to elicit, validate and document software requirements and as a consequence, obtain Software Requirement Specifications of better quality both in contents and syntax.

**PABRE** is a framework that proposes the use of Software

Requirement Pattern (SRP) as the artifact to reuse requirements knowledge. These patterns follow the typical contextproblem-

solution scenario of patterns, basically consists of: a template that may generate one or more requirements by tailoring the template (solution) to a certain project, and some information (context-problem) to identify its need to be applied in that particular project.

This document is a **excerpt** the requirements specification document of the **PABRE-MAN system**, that allows the creation and management of SRP, and that provides access to these SRP through the **PABRE-WS web services subsystem**. SRP are patterns that facilitate the definition of requirements that are used in multiple systems with small variations.

The PABRE-MAN system scope is limited to the definition of the SRP but it does not cover the use of these SRP during the elicitation of the software requirements of a system. In order to use the SRP another software system is required: the desktop tool PABRE-PROJ or by the Chrome App PABRE-RW.

*/\* This document just includes an **excerpt** of the requirements of the PABRE-MAN system defined by applying SRP defined in the PABRE SRP catalogue \*/*

## FUNCTIONAL REQUIREMENTS

## 1.- Functional Suitability

## NON-FUNCTIONAL REQUIREMENTS

## 2.- Performance Efficiency

- The system shall be able to manage an approximate volume of **1000 SRP**.
- The system shall response to user interface actions in **0.1 second** at most.

## 3.- Compatibility

- The system shall offer functionalities for importing and exporting data.
- The system shall be able to **export** data in **pdf, rtf, xml** formats.
- The system shall provide Application Program Interfaces (APIs) to be used with the following technologies: **REST web services that provide access to the defined SRP**.

## 4.- Usability

- The supplier shall provide automatic installation procedures over the chosen platform for **PABRE-MAN**.
- The system shall provide the user interface available in **English**.
- Users with profile of **Requirement Analyst** shall learn to use fluidly SRP definition functionalities of the user interface in maximum **1 hour**.

## 5.- Reliability

- The system shall have a minimal availability rate of **95%**.
- The system shall operate continuously **24 hours** per day and **7 days** per week.

## 6.- Security

- The system shall authenticate users.
- The authentication process shall be based on the **username and password** authentication technology.
- The system shall drop off inactive connections

## 7.- Maintainability

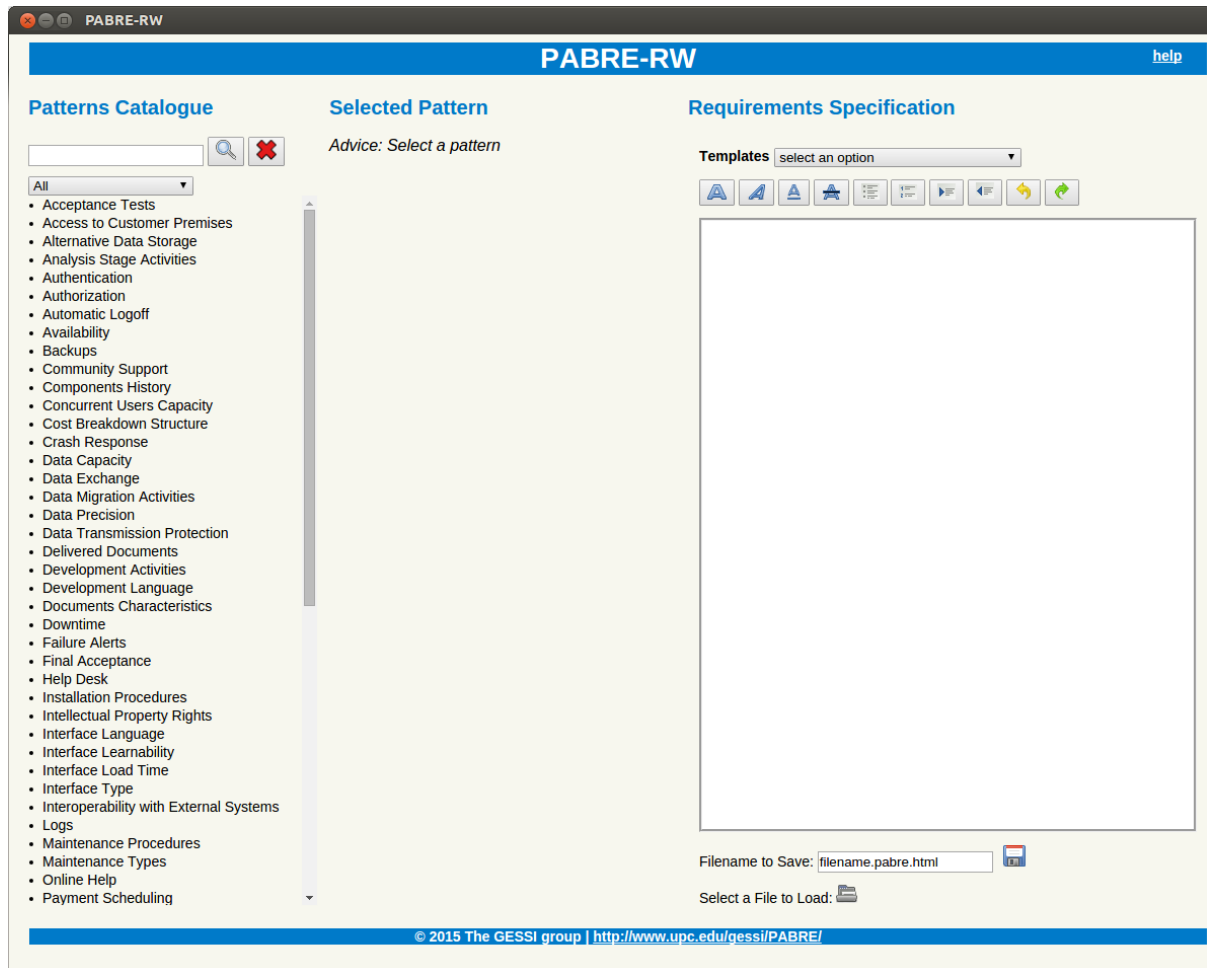
## 8.- Portability

- The **PABRE-MAN sytem** shall be developed using the **Java** development language.
- The system shall require for its operation the **PostgreSQL relational database system**
- The **web services** shall operate using **REST web services**.

Figura 41: PABRE-MAN-RequirementsExample

## 10.2 – Interfície

Aquesta és l'última versió del meu projecte, on la interfície final queda de la següent manera:



*Figura 42: Interfície final*

## 11 – Conclusions











Un cop realitzat el projecte, he arribat a les següents conclusions:

- Les eines utilitzades necessiten un temps d'aprenentatge però són bastant *senzilles* d'utilitzar. Hi ha una desavantatge i és que les Apps de Chrome no tenen tots els mètodes activats i cal fer uns ajustaments més complicats.
- Aquest projecte, a més de poder finalitzar el meu grau, m'ha permès *aprendre* molt. He posat en pràctica el coneixement que ja tenia per a fer projectes de software, i he après noves tecnologies.
- D'aquest projecte s'ha portat un *seguiment* durant tot el trimestre amb la Carme Quer (directora del projecte) i la Cristina Palomares que m'han ajudat sempre que he tingut algun problema. També m'han proporcionat una *guia* i un *suport* fonamentals per a desenvolupar la meva tasca.
- He après el funcionament de *PABRE-WS*, n'he entès l'estructura i he esbrinat com poder afegir les dades de PABRE a la meva aplicació.
- La *interfície* final és bastant senzilla. Com bé s'ha comentat en la introducció, es volia una cosa molt ergonòmica i simple per l'usuari. En aquest sentit crec que s'ha produït un producte enormement útil per als enginyers de software quan hagin de treballar amb requisits.
- Aquesta versió pot donar lloc a *possibles ampliacions*:
  1. Millorar l'escriptura d'un fitxer a disc, de manera que l'usuari pogués escollir el directori on el vol deixar.
  2. En aquesta versió només s'usa un esquema i d'aquest només el primer nivell (classificador). Fora bò que es puguin seleccionar tots els esquemes disponibles a PABRE-WS i tots els classificadors que contenen.
  3. Millorar l'editor de text perquè sigui més enriquit: justificació, tipus i grandària de caràcter, colors, ...
  4. Facilitar la substitució de les variables dins del editor pels suggeriments de la manera més automàtica i senzilla possible.
  5. Millorar l'ajuda perquè sigui més completa i interactiva: HTML en comptes de pdf, index si pot ser a l'esquerra fixe, hiperlinks interns i potser externs (Google, PABRE, etc), cerca interna de conceptes.
  6. Afegir més templates d'exemple de projecte per les aplicacions més habituals (nòmines, comptabilitat, etc.) .
  7. L'aplicació pugui guardar els projectes de manera *no local*, ja sigui en el servidor de PABRE o en el servidor de Google Storage.
  8. Exportar el projecte editat a format estàndard, ja sigui HTML estricte i complet, pdf, l'estàndard ISO odt, Microsoft Word o altres.
  9. També s'ha de tenir en compte les dependències entre patrons, de manera que si l'usuari posa un patró, l'app refereixi automàticament als patrons que tenen dependència amb el patró posat a l'editor.




10. L'aplicació dóna molta llibertat al usuari alhora de posar qualsevol tipus de part, és a dir, o parts fixes o parts esteses. Es vol millorar això, fent que si el usuari posa un part estesa d'un form d'un patró, també hagi de posar la part fixa d'aquell form.
11. També es voldria millorar el guardar projectes de manera que es pogués compartir un projecte amb altres usuaris amb Google Drive.

En la següent taula es mostra l'assoliment dels objectius previstos:

Objectiu	Assolit
<b>Objectiu 1:</b> Comprovar que es pot accedir als SRPs de la base de dades de PABRE mitjançant els serveis web de PABRE (PABRE-WS). Entendre l'estructura d'un SRP i l'estructura dels documents que retornen els serveis web. Entendre els elements d'un SRP i com s'usen els SRP en la definició de requisits.	
<b>Objectiu 2:</b> Estudiar la documentació existent sobre com desenvolupar aplicacions de Chrome. Escollir el software més adequat per al desenvolupament de l'aplicació. Analitzar la millor manera d'estructurar l'aplicació que es vol desenvolupar.	
<b>Objectiu 3:</b> Desenvolupar una primera versió de PABRE-RW que permeti escriure requisits, guardar aquests requisits en una base de dades local.	
<b>Objectiu 4:</b> a) Desenvolupar una segona versió de PABRE-RW que permeti recuperar els requisits d'una base de dades local, editar-los ... b) ... i guardar-los altre cop modificats.	 
<b>Objectiu 5:</b> Crear una tercera versió de PABRE-RW on s'afegeixi una nova àrea en la que es mostrin els SRPs i el contingut de cada un d'ells.	
<b>Objectiu 6:</b> Crear una quarta versió de PABRE-RW on s'afegeixin facilitats de cerca/ordenació dels SRPs i del contingut de cada un d'ells.	
<b>Objectiu 7:</b> Crear una cinquena versió de PABRE-RW on es pugui seleccionar una part d'un SRP i usar-la en la definició de requisits.	
<b>Objectiu 8:</b> Crear una sisena versió de PABRE-RW on s'afegeixi la cerca d'un esquema ( <i>Classifier Schema</i> ), que és un sistema jeràrquic de classificació dels SRPs.	
<b>Objectiu 9:</b> Crear una setena versió de PABRE-RW on es puguin seleccionar uns projectes ja predeterminats que poden servir d'exemple.	

Taula 12: Objectius assolits

 **Objectiu 4-b:** Es va voler millorar l'objectiu de guardar requisits, de manera que es pogués seleccionar en quina carpeta de l'ordinador es volia guardar el fitxer. No s'ha pogut trobar solució i es mostra un pop-up amb l'explicació d'on es guarda.

## 12 – Glossari

**App de Chrome:** Les apps de Chrome són unes aplicacions d'escriptori en el sentit que funcionen localment i poden estar en una finestra pròpia. Cal, però, tenir el Chrome instal·lat i funcionant com a procés actiu, ja que l'App en depèn.

**BD:** Base de Dades

**Classifier Schemas:** PABRE-WS té diversos sistemes de classificació jeràrquica que són els esquemes (classification schemas). Aquests tracten per facilitar la comprensió i la reutilització durant el procés d'obtenció, els patrons s'ordenen en un catàleg per seguir uns esquemes jeràrquics de classificació.

**CoffeeScript:** és un llenguatge que compila a Javascript.

**CSS:** Cascading Style Sheets, és un llenguatge per poder dissenyar l'aparença visual del document estructurat per HTML i XML.

**Dart:** és un llenguatge de codi obert desenvolupat per Google. No pretén substituir a JavaScript, sinó millorar-lo.

**Div:** Element de HTML que s'utilitza per fer una divisió o secció en el document.

**Enterprise Architect:** eina de disseny i modelatge visual per a sistemes de software que, entre altres coses, permet gestionar els requisits.

**ESSI:** Enginyeria del Software i els Serveis, departament de la UPC

**Form:** Element d'un SRP que està compost per diverses *parts* que són les que contenen els *requisits*. A cada form sempre hi ha una part anomenada part fixe (*fixed*), i hi pot haver altres parts anomenades esteses (*extended*).

**Gantt:** els diagrames de Gantt mostren la planificació del desenvolupament.

**GEP:** Gestió de Projectes

**GESSI:** Grup d'Enginyeria del Software i dels Serveis, grup d'investigació enquadrat en la UPC

**Html:** HyperText Markup Language, és un llenguatge estàndard per a l'estructura bàsica d'una pàgina web.

**HTTP:** Hypertext Transfer Protocol, és un protocol per a la comunicació usat en Internet i que pot suportar, entre altres coses, el HTML.

**JavaScript:** és un llenguatge de programació orientat a objectes per a desenvolupar programes web.

**JSON:** JavaScript Object Notation, subconjunt de la notació literal d'objectes de JavaScript que no requereix XML.

**Mètriques / Metrics:** Cada mètrica s'uneix a un paràmetre que està unit a una part fixa o part estesa d'un form d'un SRP, és a dir, a una variable que té una part. Les mètriques poden enumerar valors (per exemple, noms de plataformes middleware), sencer (per exemple, per indicant el nombre de connexions amb el suport), els nombres reals (per exemple, per al mesurament de temps de resposta) i valors booleans (per exemple, per saber si algun protocol és compatible).

**MySQL:** és una base de dades relacional.

**PABRE:** sistema de gestió de SRP administrat per GESSI. Conté PABRE.MAn, PABRE.Proj i PABRE-WS

**PABRE-Man:** subsistema de PABRE que permet crear nous SRP i afegir-los a la base de dades.

**PABRE-Proj:** subsistema de PABRE que permet escriure requisits basats en els SRP.

**PABRE-WS:** Són serveis web REST de PABRE que permeten a altres sistemes externs a PABRE accedir a la base de dades de SRP.

**Part fixa o part estesa:** corresponen diversos refinaments del patró, defineixen i concreten les accions o mètodes que portaran a complir l'objectiu global.

**Patró / Pattern:** els patrons de requisits són generalitzacions estandarditzades que contenen organitzadament models de requisits. Cada patró ha de tenir unes característiques que cal complir.

**Requisits / Requirements:** un requisit és una descripció per marcar una exigència que ha de complir el sistema software.

**ReqView:** app de Chrome per gestionar requisits.

**Scrum:** és una metodologia àgil per desenvolupar projectes.

**Servei web REST:** (Representational State Transfer) és un estil d'arquitectura de programari que es fa servir en aplicacions de xarxa distribuïdes, típicament sobre HTTP com és el nostre cas.

**SRP:** Software Requirement Pattern, patró de requisits de software

**SRPs:** catàleg de patrons de requisits de software.

**Storage:** Base de dades

**Textarea:** és un element de HTML. Aquest element és una entrada de text de diverses línies que es poden editar.

**TFG:** Treball Final de Grau

**UML:** Unified Modeling Language, llenguatge estàndard per a anàlisi, disseny i modelatge de software.

**UPC:** Universitat Politècnica de Catalunya.

**URI:** Uniform Resource Identifier.

**WebStorm:** Aplicació d'entorn de programació per desenvolupar aplicacions web.

## 13 – Referències

[1] <https://developer.chrome.com/apps/>

Aquesta és la web oficial de desenvolupadors de Google Chrome per aplicacions. Conté tota la informació necessària per a desenvolupar apps. Aquesta pàgina redirigirà a altres per a cada tema especialitzat.

[2] <https://developer.chrome.com/apps/storage>

Aquesta web forma part de la web oficial de desenvolupadors de Google Chrome per a apps i està especialitzada en emmagatzemament de dades.

[3] <http://www.capterra.com/requirements-management-software/>

Aquesta web fa referència a les diferents aplicacions que existeixen per treballar amb requisits.

[4] <https://chrome.google.com/webstore/detail/reqview-desktop/cmbnhmeabefpflldnahpbghcejkbli?hl=es>

Aquesta és la web per instal·lar l'aplicació Req View de Google Chrome que existeix per treballar requisits.

[5] <http://docs.pabrews.apiary.io/>

Aquesta és la web per accedir a la base de dades PABRE.

[6] <http://www.fib.upc.edu/fib/estudiar-enginyeria-informatica/assignatures.html?assign=ER>

Aquesta és la web on es descriu l'Enginyeria de Requisits.

[7] [https://support.google.com/chrome\\_webstore/answer/1047776?hl=ca](https://support.google.com/chrome_webstore/answer/1047776?hl=ca)

Aquesta web fa referència a l'explicació que és Chrome Web Store i com funciona.

[8] <http://www.scrum.org/>

Aquesta és la web oficial de la metodologia àgil Scrum.

[9] <https://www.inflectra.com/SpiraTest/Highlights.aspx#Bug-Tracking>

[10] <https://www.process.st/product/>

[11] <https://www.tracecloud.com/GloreeJava2/jsp/WebSite/TCHome.jsp>

[12] <http://www.gatherspace.com/>

[13] <http://www.requirementone.com/>

- [14] <http://www-01.ibm.com/software/es/rational/>
- [15] <https://www.innoslate.com/>
- [16] <https://www.jamasoftware.com/>
- [17] <http://www.sparxsystems.com.au/products/ea/>
- [18] <https://www.reqview.com/>
- [19] <http://www.upc.edu/gessi/PABRE/Tools.html>
- [20] <https://ca.wikipedia.org/wiki/REST>
- [21] Fernando Mora Bernabé  
*Development of web services for the PABRE system*  
Master's Degree in Information Technology (MTI)
- [22] <http://pabre.essi.upc.edu/ws/api/patterns>
- [23] <http://docs.pabrews.apiary.io/>
- [24] <https://api.dartlang.org/1.13.0/dart-html/HttpRequest-class.html>
- [25] [https://developer.chrome.com/apps/first\\_app](https://developer.chrome.com/apps/first_app)
- [26] <http://www.w3schools.com/js/>
- [27] <https://www.dartlang.org/>
- [28] <http://coffeescript.org/>
- [29] <https://es.wikipedia.org/wiki/Dart>
- [30] <https://www.dartlang.org/downloads/>
- [31] <https://chrome.google.com/webstore/detail/chrome-apps-extensions-de/ohmmkhmmmpcnpikjeljgnaoabkaalbgc>
- [32] <https://www.dartlang.org/tools/pub/dart2js-transformer.html>
- [33] <https://www.jetbrains.com/webstorm/download/#section=linux-version>

## Annex: Índex de taules

Taula 1: Aplicacions de gestions de requisits.....	12
Taula 2: Hores dedicades.....	19
Taula 3: Rols i els preus de cada rol.....	22
Taula 4: Pressupost recursos humans.....	22
Taula 5: Pressupost hardware.....	23
Taula 6: Pressupost software.....	23
Taula 7: Costos de riscos.....	24
Taula 8: Costos de imprevistos.....	25
Taula 9: Pressupost total.....	25
Taula 10: Sostenibilitat.....	27
Taula 11: patró Failure Alerts.....	29-31
Taula 12: Objectius assolits.....	72

## Annex: Índex de figures

Figura 1: Disseny aplicació .....	9
Figura 2: Disseny del sistema PABRE .....	10
Figura 3: Base de dades dels requisits d'un projecte .....	14
Figura 4: Fase inicial .....	20
Figura 5: Desenvolupament.....	20
Figura 6: Fase final .....	20
Figura 7: Estructura bàsica dels SRPs .....	28
Figura 8: estructura bàsica del patró Failure Alerts .....	29
Figura 9: Esquemes del sistema PABRE .....	33
Figura 10: Diagrama UML .....	34
Figura 11: Esquema de l'API .....	35
Figura 12: Casos d'ús .....	38
Figura 13: app de Chrome .....	44
Figura 14: manifest.json de PABRE-RW .....	45
Figura 15: background.json de PABRE-RW .....	45
Figura 16: app de Chrome amb Dart .....	46
Figura 17: Nou projecte .....	47
Figura 18: Selecció de tipus de projecte .....	47
Figura 19: Imatge per baixar l'app "Chrome Apps & Extensions Developer Tool" ..	48
Figura 20: Icona de l'app "Chrome Apps & Extensions Developer Tool" .....	48
Figura 21: pubspec.yaml .....	49
Figura 22: Carpetes i fitxers dintre del directori del projecte .....	49
Figura 23: Carpetes i fitxers dintre del directori web .....	50
Figura 24: Carpetes i fitxers dintre del directori build .....	50
Figura 25: Disseny bàsic de la interfície .....	51
Figura 26: Disseny quan s'engega la interfície .....	52
Figura 27: Disseny de la interfície seleccionant un patró .....	53
Figura 28: Disseny de la interfície seleccionant un form .....	54
Figura 29: Disseny de la interfície seleccionant una part .....	55
Figura 30: Icones de l'editor de text .....	57
Figura 31: Interfície de l'editor de text .....	59
Figura 32: Comparació dels botons dels noms dels SRPs .....	61
Figura 33: Llista dels SRPs .....	62
Figura 34: Selecció d'un SRP i mostrar els froms .....	62
Figura 35: Selecció d'un from i mostrar les parts .....	63
Figura 36: Cerca de paraules .....	64
Figura 37: Llista de les parts .....	65
Figura 38: Selecció d'una part .....	66
Figura 39: Selecció d'esquema .....	67
Figura 40: Iso25010schema .....	68-69
Figura 41: PABRE-MAN-RequirementsExample .....	69
Figura 42: Interfície final .....	70



## Annex: Anàlisi

### 1 – Cas d'ús: Editar requisits

#### Descripció

L'usuari vol iniciar l'app.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

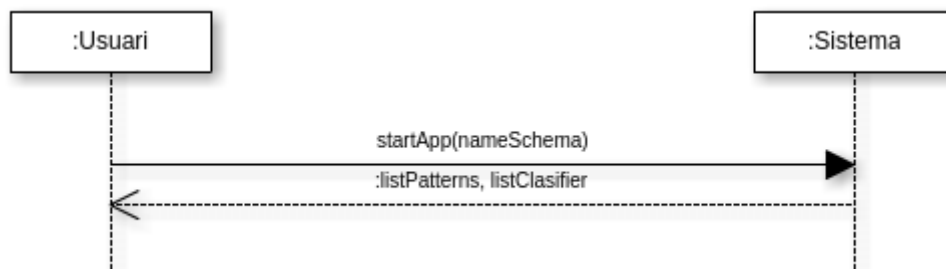
**Disparador:** L'usuari vol executar l'app.

**Escenari d'èxit principal:**

1. L'usuari inicialitza l'app.
2. El sistema mostra els noms dels patrons i els noms dels classificadors.
3. S'acaba el cas d'ús.

**Extensions:** -

#### Diagrama de seqüència



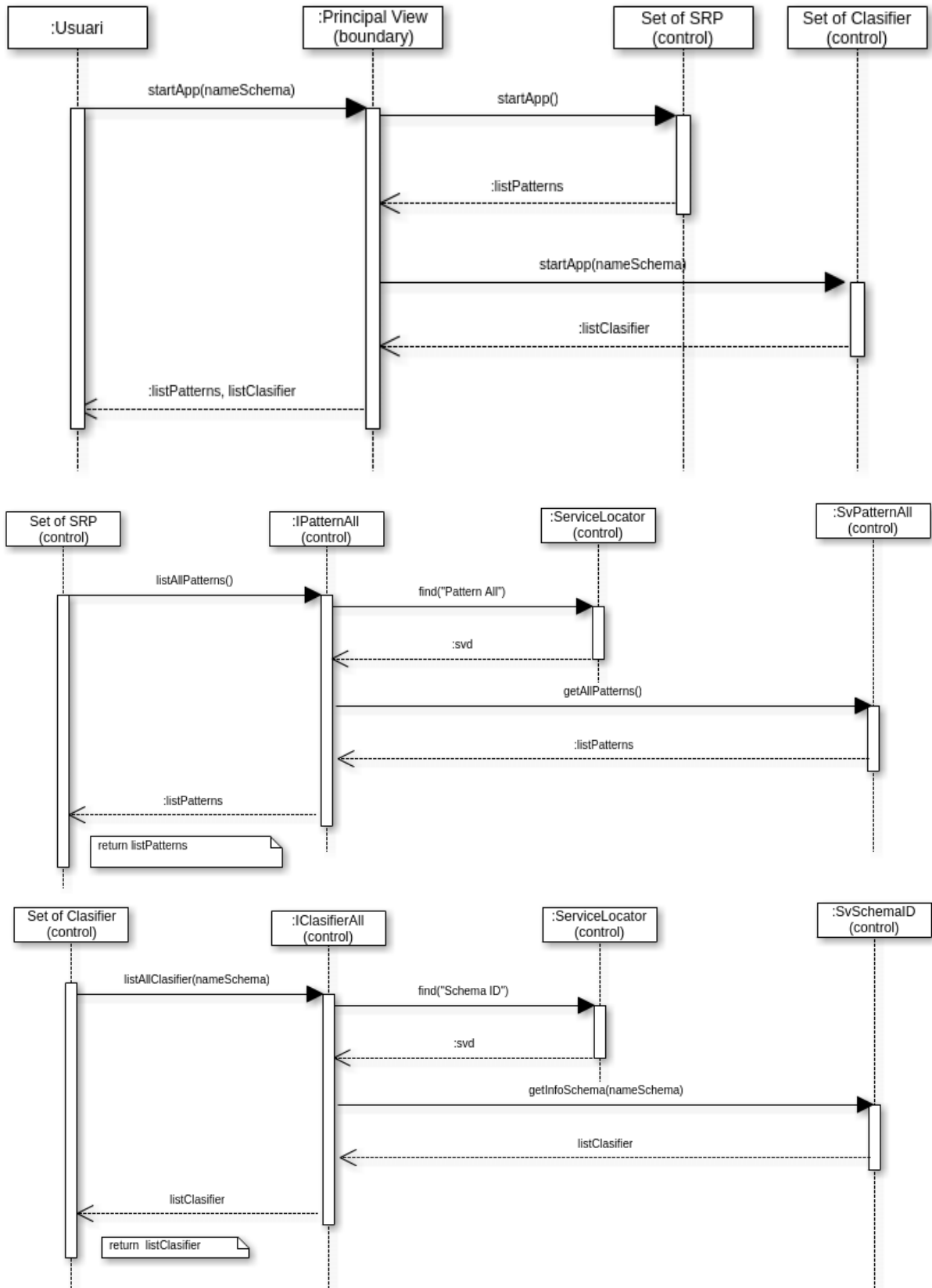
#### Contracte de l'operació.

#startApp(nameSchema): llistPatterns, listClasifier

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol iniciar l'app.

**pre:** -

**post:** retorna la llista dels patrons i la llista dels classificadors d'un esquema.

**Operació: startApp(nameSchema): llistPatterns, listClasifier***Diagrama de seqüència de l'operació startApp*

## 2 – Cas d'ús: Editar requisits

### Descripció

L'usuari vol poder escriure i editar requisits. En aquest cas d'ús l'usuari podrà modificar els seus requisits.

### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol editar els requisits.

**Escenari d'èxit principal:**

3. L'usuari gestiona la informació en l'editor de text.
4. S'acaba el cas d'ús.

**Extensions:**

- 1a. L'usuari usa l'editor de text.
- 1b. L'usuari vol canviar la negreta
  - 1b1. L'usuari selecciona la negreta
  - 1b2 El sistema activa/desactiva la negreta en l'editor
- 1c. L'usuari vol canviar la cursiva
  - 1c1. L'usuari selecciona la cursiva
  - 1c2 El sistema activa/desactiva la cursiva en l'editor
- 1d. L'usuari vol canviar el subratllat
  - 1d1. L'usuari selecciona el subratllat
  - 1d2 El sistema activa/desactiva el subratllat en l'editor
- 1e. L'usuari vol canviar el ratllat
  - 1e1. L'usuari selecciona el ratllat
  - 1e2 El sistema activa/desactiva el ratllat en l'editor
- 1f. L'usuari vol canviar la llista de punts
  - 1f1. L'usuari selecciona la llista de punts
  - 1f2 El sistema activa/desactiva la llista de punts en l'editor
- 1g. L'usuari vol canviar la llista de números
  - 1g1. L'usuari selecciona la llista de números
  - 1g2 El sistema activa/desactiva la llista de números en l'editor
- 1h. L'usuari vol canviar la indentació
  - 1h1. L'usuari selecciona la indentació
  - 1h2 El sistema afegeix una indentació en l'editor
- 1i. L'usuari vol canviar la desindentació
  - 1i1. L'usuari selecciona la desindentació
  - 1i2 El sistema afegeix una desindentació en l'editor
- 1j. L'usuari vol anar cap enrere
  - 1j1. L'usuari selecciona l'enrere
  - 1j2 El sistema canvia la informació de l'editor per la que hi havia anteriorment
- 1k. L'usuari vol anar cap endavant
  - 1k1. L'usuari selecciona l'endavant

1k2 El sistema canvia la informació de l'editor per la que hi havia posteriorment.

### Diagrama de seqüència

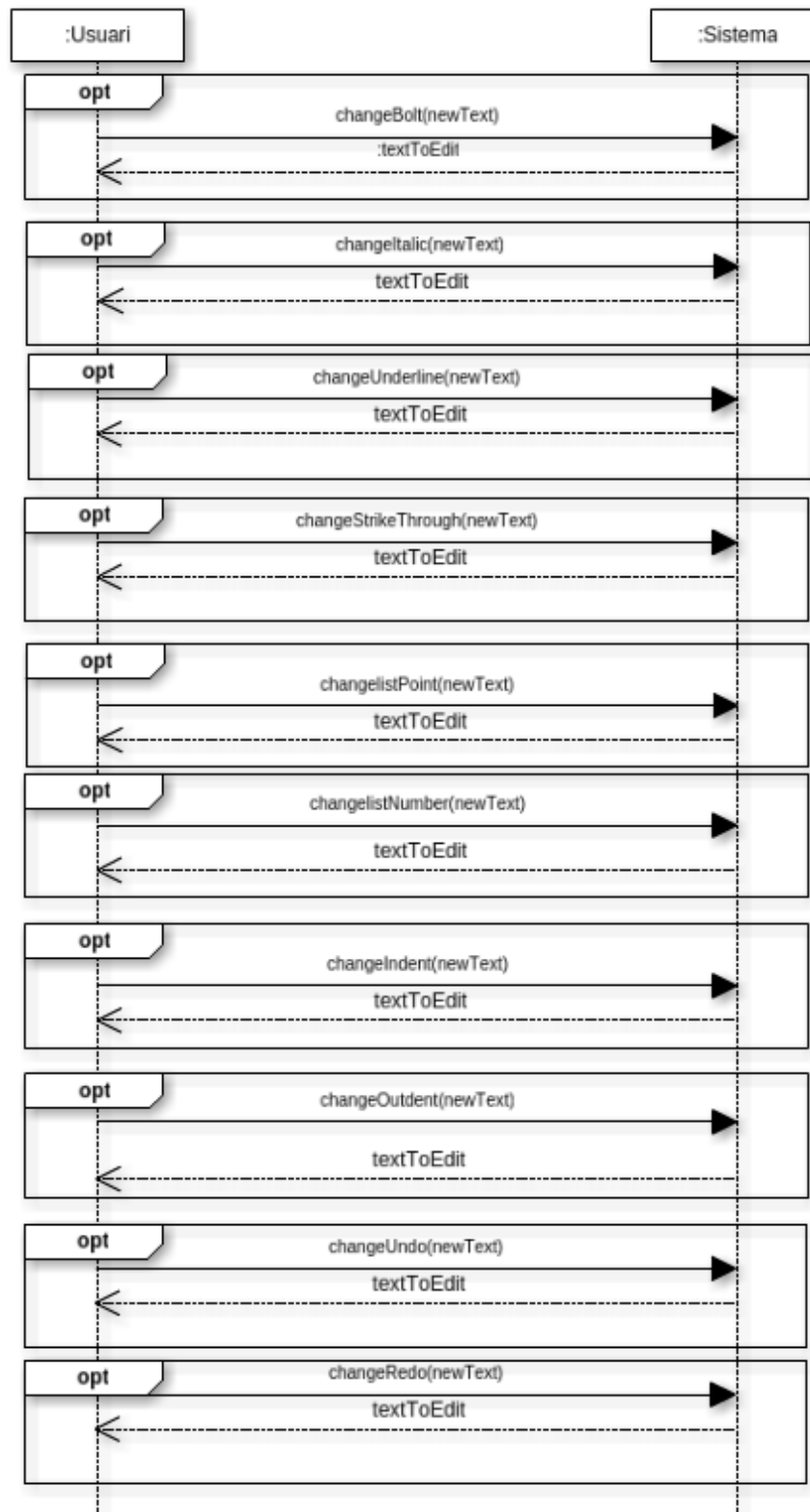


Diagrama de seqüència de cas d'ús Editar Requisits

**Contracte de l'operació.**

**#changeBolt(newText):** textToEdit

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol canviar a negreta, és a dir, activar-la o desactivar-la

**pre:** -

**post:** retorna la informació de l'editor amb la negreta activada/desactivada en l'interval seleccionat (o al voltant del cursor). S'activarà si estava alguna part de l'interval (o tot) sense negreta, es desactivarà si estava tot l'interval en negreta.

**#changeltalic(newText):** textToEdit

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol canviar a cursiva, és a dir, activar-la o desactivar-la

**pre:** -

**post:** retorna la informació de l'editor amb la cursiva activada/desactivada en l'interval seleccionat (o al voltant del cursor). S'activarà si estava alguna part de l'interval (o tot) sense cursiva, es desactivarà si estava tot l'interval en cursiva.

**#changeUnderline(newText):** textToEdit

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol canviar a subratllat, és a dir, activar-lo o desactivar-lo

**pre:** -

**post:** retorna la informació de l'editor amb el subratllat activat/desactivat en l'interval seleccionat (o al voltant del cursor). S'activarà si estava alguna part de l'interval (o tot) sense subratllat, es desactivarà si estava tot l'interval en subratllat.

**#changeStrikeThrough(newText):** textToEdit

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol canviar a ratllat, és a dir, activar-lo o desactivar-lo

**pre:** -

**post:** retorna la informació de l'editor amb el ratllat activat/desactivat en l'interval seleccionat (o al voltant del cursor). S'activarà si estava alguna part de l'interval (o tot) sense ratllat, es desactivarà si estava tot l'interval en ratllat.

**#changelistPoint(newText):** textToEdit

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol canviar a la llista de punts, és a dir, activar-la o desactivar-la

**pre:** -

**post:** retorna la informació de l'editor amb la llista de punts activada/desactivada en l'interval seleccionat (o al voltant del cursor). S'activarà si estava alguna part de l'interval (o tot) sense llista de punts, es desactivarà si estava tot l'interval en llista de punts.

**#changelisNumber(newText):** textToEdit

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol canviar a la llista de números, és a dir, activar-la o desactivar-la

**pre:** -

**post:** retorna la informació de l'editor amb la llista de números activada/desactivada en l'interval seleccionat (o al voltant del cursor). S'activarà si estava alguna part de l'interval (o tot) sense llista de números, es desactivarà si estava tot l'interval en llista de números.

**#changeIndent(newText): textToEdit**

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol afegir una indentació.

**pre:** -

**post:** retorna la informació de l'editor indentant l'interval seleccionat (o al voltant del cursor).

**#changeOutdent(newText): textToEdit**

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol canviar a la desindentació

**pre:** -

**post:** retorna la informació de l'editor desindentant l'interval seleccionat (o al voltant del cursor) en cas que hagi estat indentat.

**#changeUndo(newText): textToEdit**

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol anar enrere.

**pre:** -

**post:** retorna la informació de l'editor que hi havia anteriorment, en cas de que hi hagi informació anteriorment.

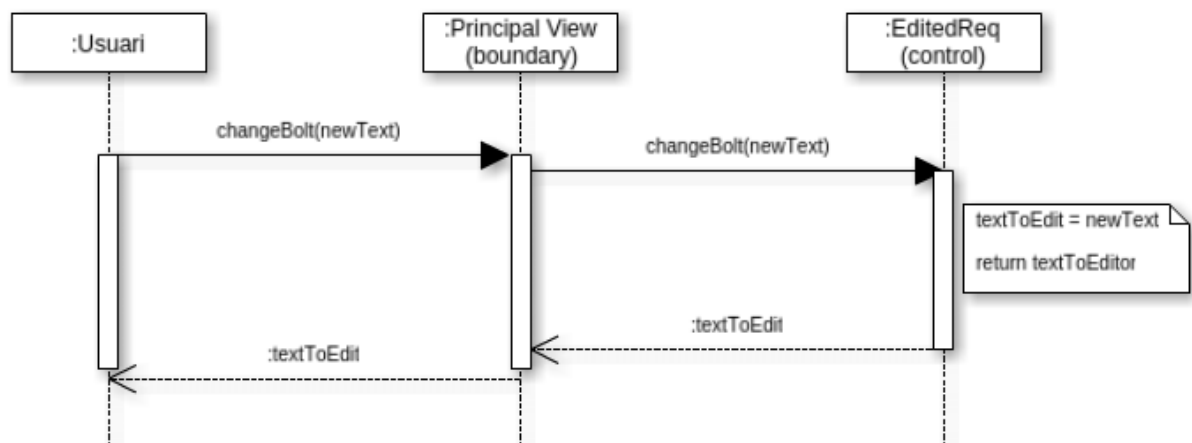
**#changeRedo(newText): textToEdit**

**Descripció:** Aquest esdeveniment s'utilitza quan l'usuari vol anar endavant.

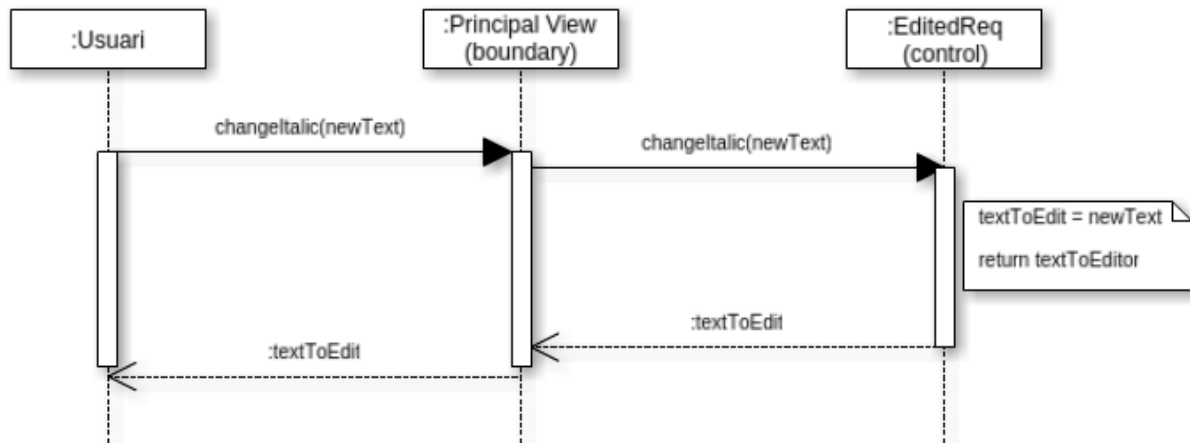
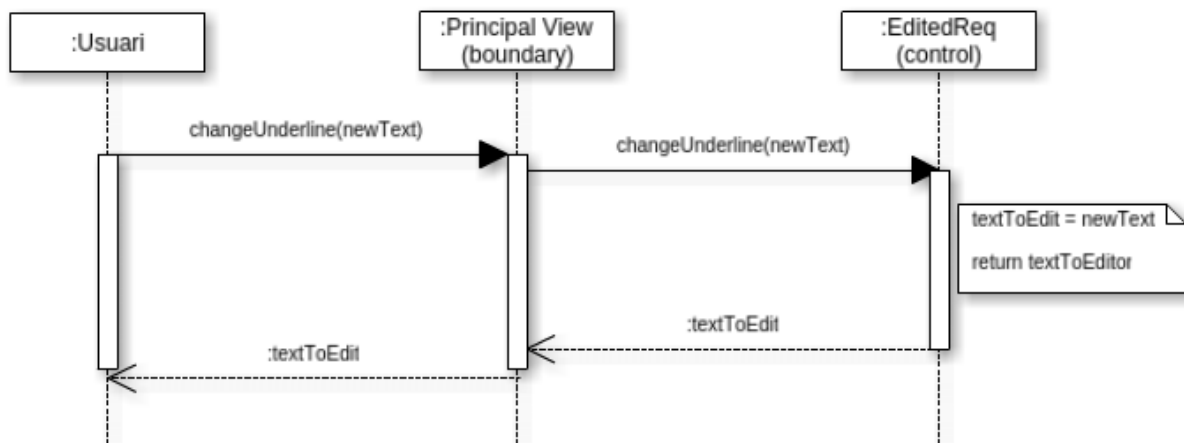
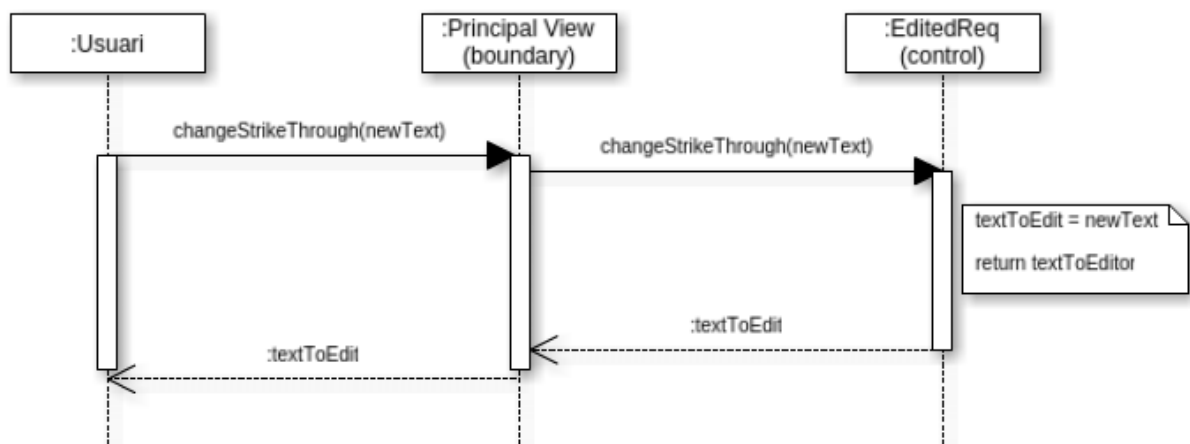
**pre:** -

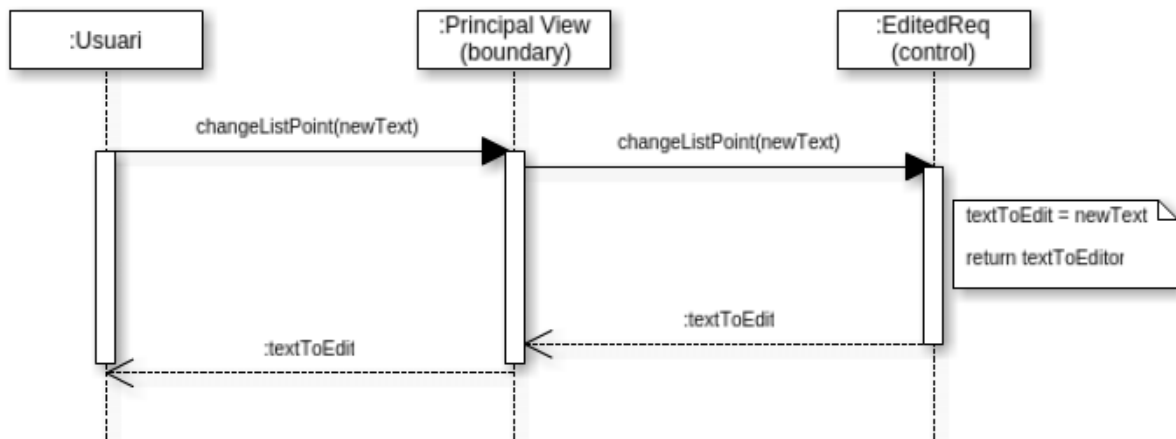
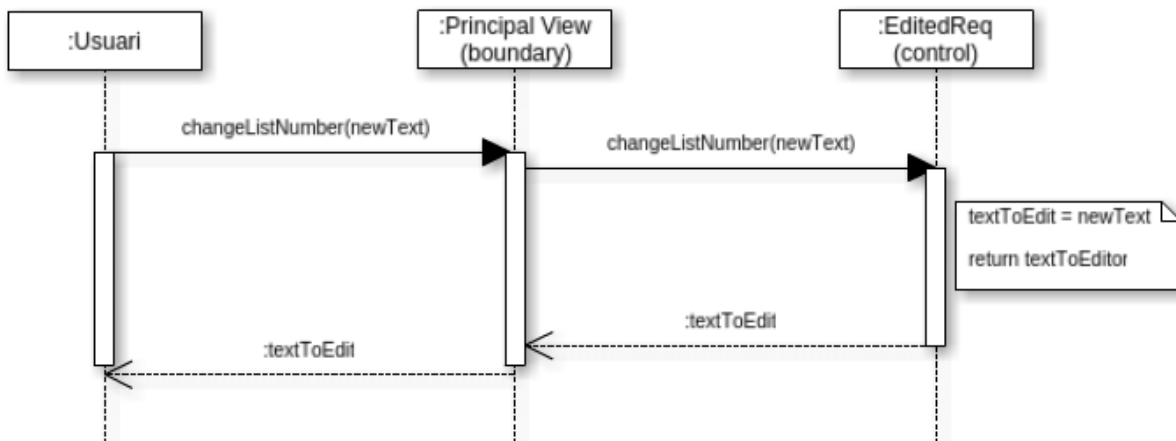
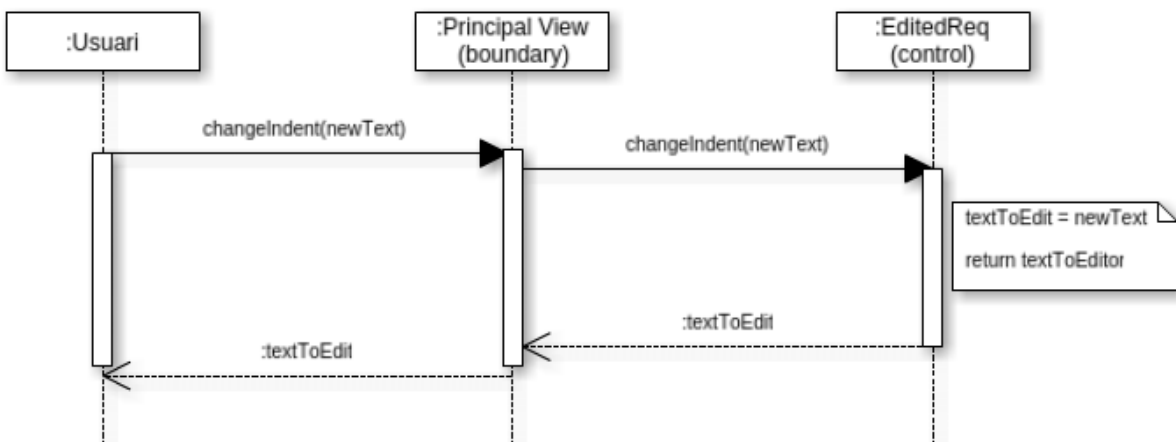
**post:** retorna la informació de l'editor que hi havia posteriorment, en cas de que hi hagi informació posteriorment.

**Operació: changeBolt(newText): textToEdit**

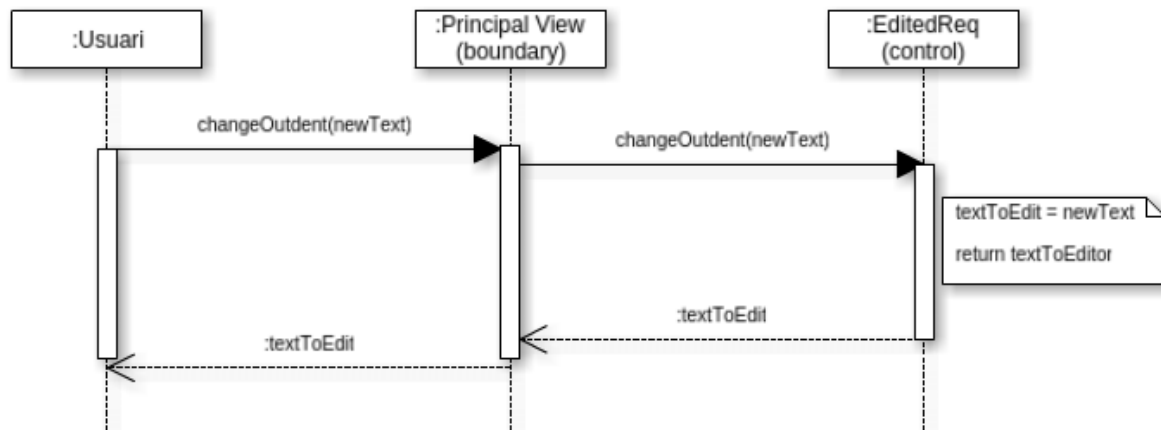
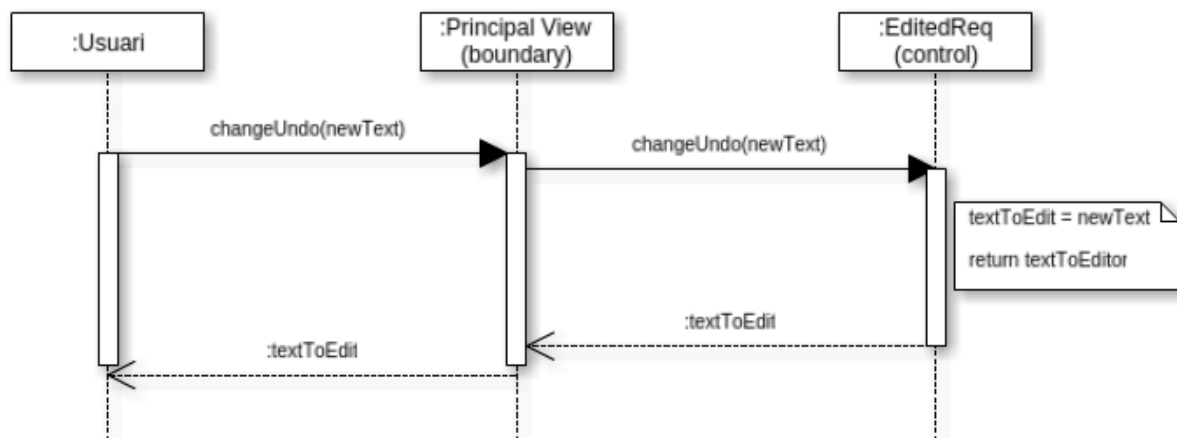
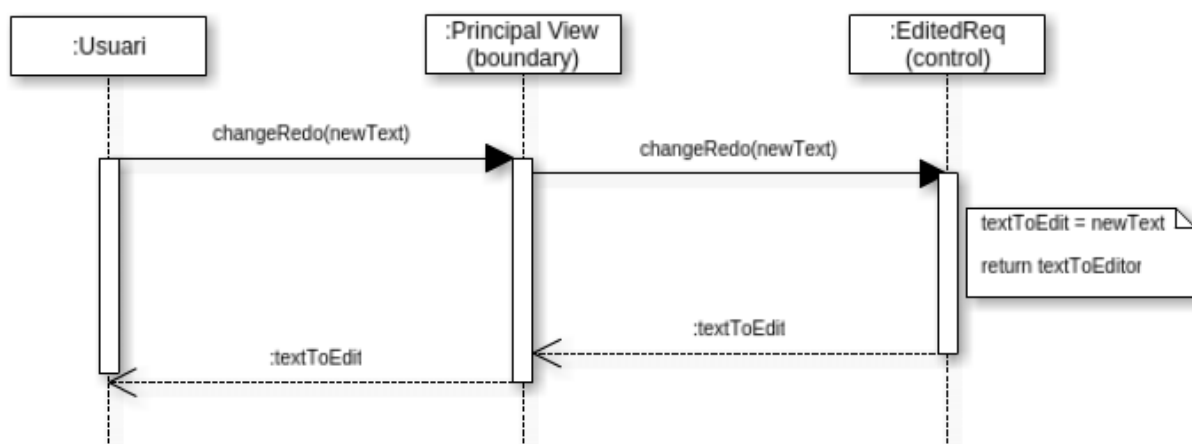


*Diagrama de seqüència de l'operació changeBolt*

**Operació: changeltalic(newText)***Diagrama de seqüència de l'operació changeltalic***Operació: changeUnderline(newText): textToEdit***Diagrama de seqüència de l'operació changeUnderline***Operació: changeStrikeThrough(newText): textToEdit***Diagrama de seqüència de l'operació changeStrikeThrough*

**Operació: changelistPoint(newText): textToEdit***Diagrama de seqüència de l'operació changelistPoint***Operació: changelisNumber(newText): textToEdit***Diagrama de seqüència de l'operació changelisNumber***Operació: changelIndent(newText): textToEdit***Diagrama de seqüència de l'operació changelIndent*



**Operació: changeOutdent(newText): textToEdit***Diagrama de seqüència de l'operació changeOutdent***Operació: changeUndo(newText): textToEdit***Diagrama de seqüència de l'operació changeUndo***Operació: changeRedo(newText): textToEdit***Diagrama de seqüència de l'operació changeRedo(newText)*

### 3 – Cas d'ús: Guardar requisits

#### Descripció

L'usuari vol poder guardar els requisits que hi ha a l'editor de l'aplicació, en un fitxer en la base de dades local.

#### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol guardar un fitxer en la base de dades local.

**Escenari d'èxit principal:**

1. L'usuari indica que vol guardar les seves dades.
2. S'ha acabat el cas d'ús.
3. El sistema obre un pop-up (una nova finestra) i mostra un missatge indicant que el fitxer s'ha guardat correctament

**Extensions:**

- 1a. L'usuari escriu el nom amb el que vol guardar el fitxer.
- 2a. L'extensió és errònia.
  - 2a1. El sistema mostra un pop-up i mostra un missatge amb l'extensió que ha de tenir el fitxer.
  - 2a2. L'usuari ha de prémer el botó per tancar el pop-up
  - 2a3. Es torna al punt 1.

#### Diagrama de seqüència



Diagrama de seqüència de cas d'ús Guardar requisits

#### Contracte de l'operació.

#downloadFile(nameFile, textFile): Message done/failed

**Descripció:** L'esdeveniment fa que es guardin els requisits d'un projecte a la base de dades de l'ordinador

**pre:** -

**post:** Message done / failed, el missatge li dirà a l'usuari si les dades s'han guardat correctament o no.

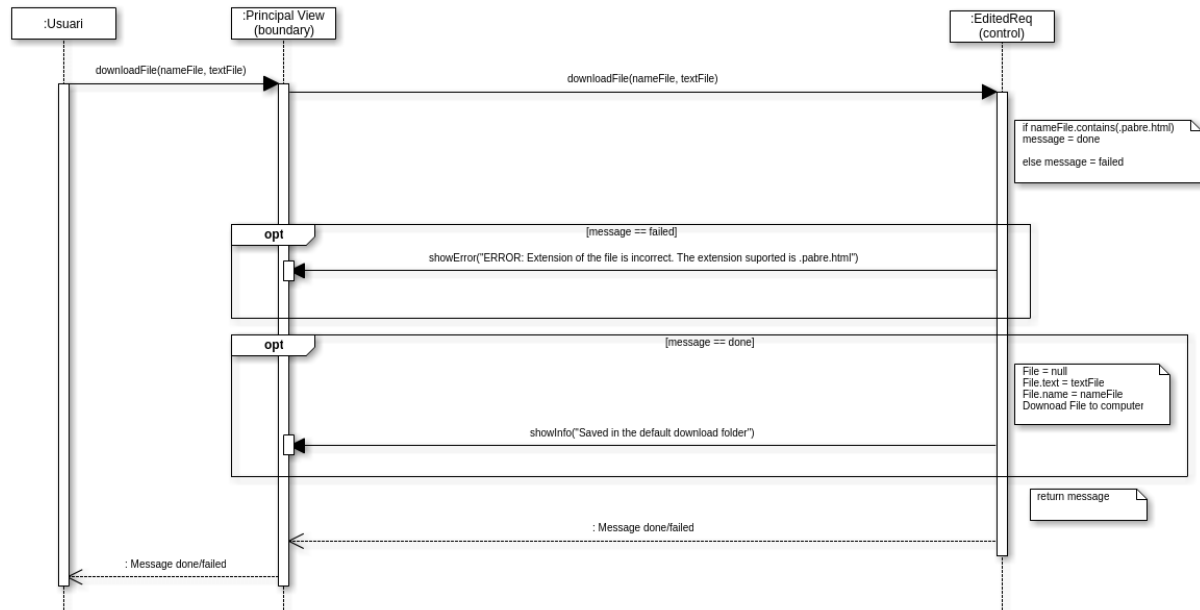
**Operació: downloadFile(nameFile, textFile): Message done/failed**

Diagrama de seqüència de l'operació downloadFile

## 4 – Cas d'ús: Recuperar requisits

### Descripció

L'usuari vol poder pujar un fitxer i mostrar-lo en l'editor de l'aplicació. Aquest fitxer està guardat a la base de dades local.

### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol pujar un fitxer a l'aplicació.

**Escenari d'èxit principal:**

1. L'usuari indica que vol pujar un fitxer.
2. El sistema obre una finestra on mostra els fitxers que té l'usuari en el seu ordinador.
3. L'usuari navega per les carpetes de la seva base de dades local fins a trobar el fitxer que vol.
4. L'usuari selecciona el fitxer que vol pujar.
5. El sistema mostra la informació que conté aquest fitxer.
6. S'acaba el cas d'ús.

**Extensions:**

- 1a. El sistema mostra una finestra amb un avís de confirmació.
  - 1a1. L'usuari ho confirma.
- 3a. L'usuari modifica el nom de les extensions que apareixen en els fitxers.
- 4a. L'extensió del fitxer és erroni

4a1. El sistema mostra un pop-up i mostra un missatge amb l'extensió que ha de tenir el fitxer

4a2. L'usuari ha de prémer el botó per treure el pop-up

4a3. Es torna al punt 1.

### Diagrama de seqüència



Diagrama de seqüència de cas d'ús recuperar requisits

### Contracte de l'operació.

#newFileLoad(nameFile, textFile): textToFile

**Descripció:** L'esdeveniment fa que es pugui un fitxer i es mostri en el àrea de l'editor de text.

**pre:** -

**post:** textToEdit és el contingut que hi havia en el fitxer i ara es mostra en l'editor de text.

### Operació: newFileLoad(nameFile, textFile): textToFile

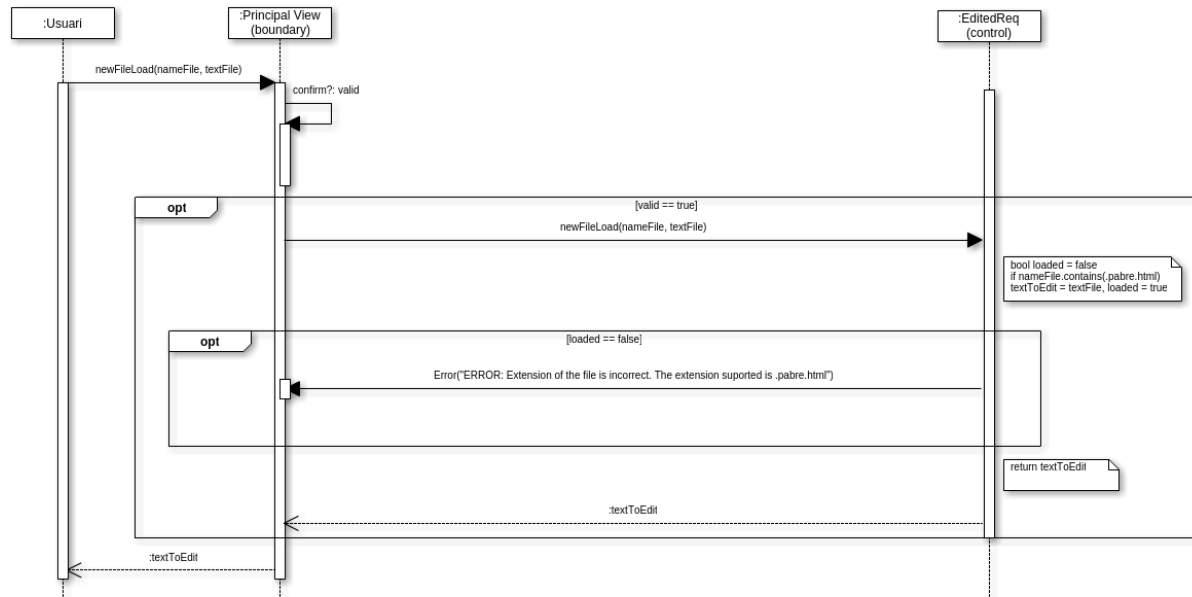


Diagrama de seqüència de l'operació newFileLoad

## 5 – Cas d'ús: Accedir als SRPs

### Descripció

L'usuari vol poder veure tota la informació que conté un patró, és a dir, la descripció del patró, els form, la descripció de cada form, les fixed i extended part.

### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari poder veure tota la informació que conté un patró.

**Escenari d'èxit principal:**

1. El sistema mostra els noms dels patrons.
2. L'usuari selecciona el patró que vol.
3. El sistema mostra informació del patró.
4. L'usuari selecciona una form.
5. El sistema mostra informació de la form seleccionada.
6. S'activa el cas d'ús "Insertar una part a l'editor"
7. S'acaba el cas d'ús.

**Extensions:** -

### Diagrama de seqüència

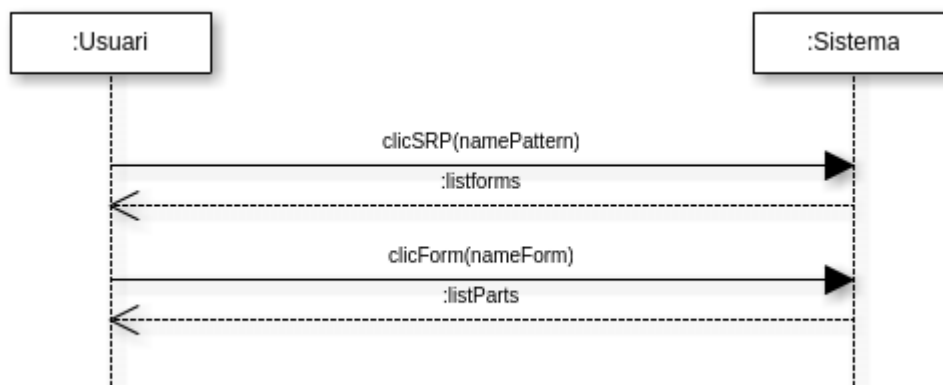


Diagrama de seqüència de cas d'ús Accedir als SRPs

### Contracte de l'operació.

#clicSRP(namePattern): listform

**Descripció:** L'esdeveniment mostra els forms que te un SRP seleccionat.

**Pre:** -

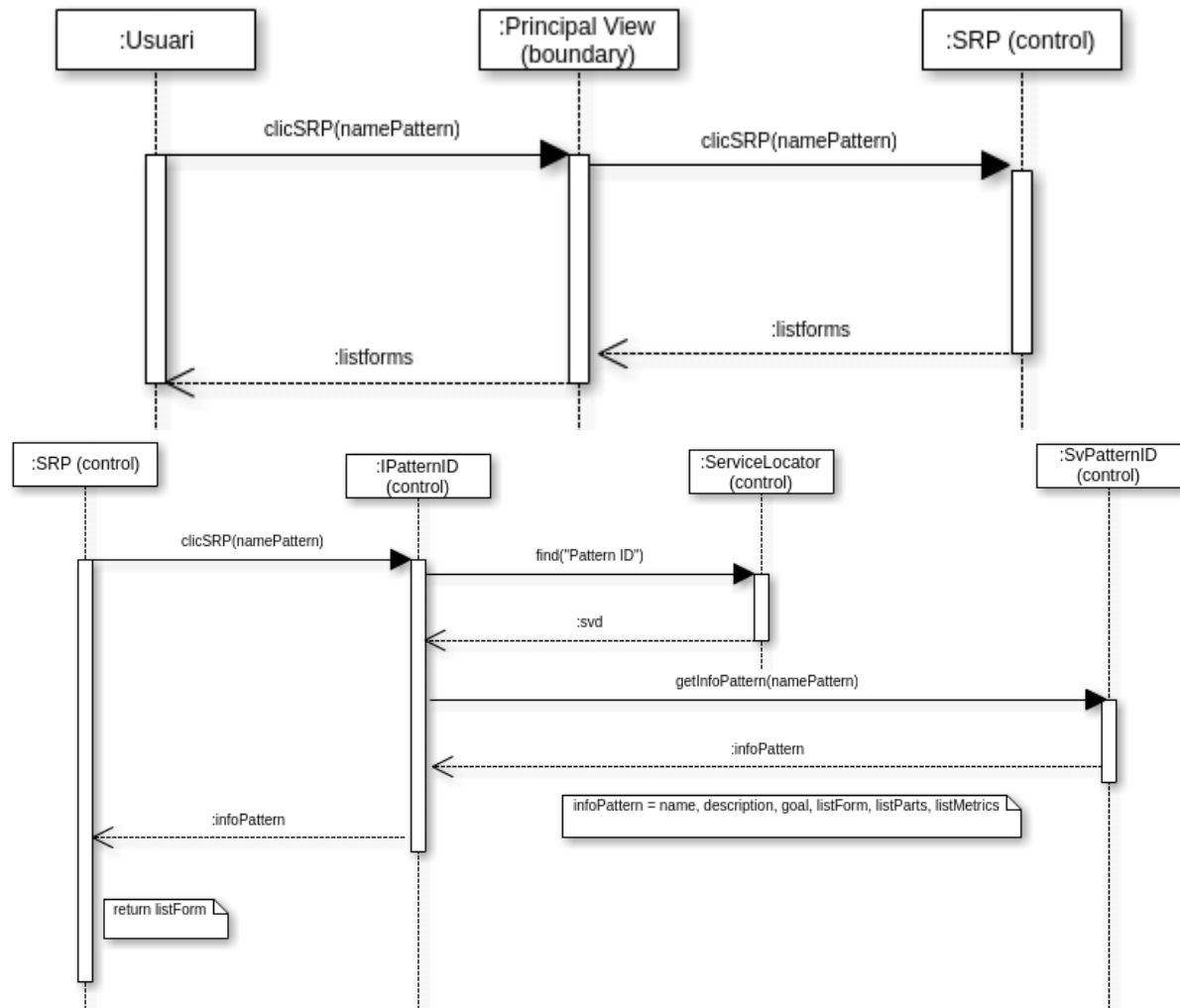
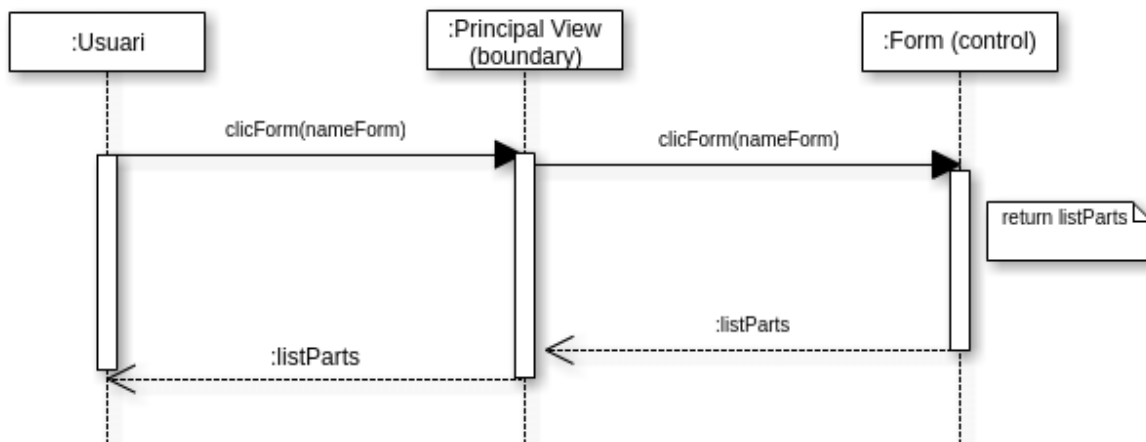
**post:** listForms. L'esdeveniment retorna la llista dels formularis que té aquell SRPs

#clicForm(nameForm): listParts

**Descripció:** L'esdeveniment mostra les parts que te un SRP.

**Pre:** -

**post:** listForms. L'esdeveniment retorna la llista de les parts que té aquell form seleccionat d'un SRPs

**Operació: clicSRP(namePattern): listform***Diagrama de seqüència de l'operació clicSRP***Operació: clicForm(nameForm): listParts***Diagrama de seqüència de l'operació clicForm*

## 6 – Cas d'ús: Insertar una part a l'editor

### Descripció

L'usuari vol poder escriure una part d'un patró a l'editor de text.

### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol poder escriure una part d'un patró a l'editor de text.

**Escenari d'èxit principal:**

1. L'usuari selecciona la part d'un patró
2. El sistema mostra la informació de les mètriques.
3. El sistema afegeix la part del patró a l'editor de text
4. S'acaba el cas d'ús.

**Extensions:**

- 1a. L'usuari vol veure les mètriques
  - 1a1 L'usuari selecciona un part
  - 1a2. El sistema mostra les mètriques
- 1b. L'usuari vol afegir una part a l'editor
  - 1b1 L'usuari selecciona el botó per afegir una part a l'editor
  - 1b2. El sistema mostra la part, el requisit, afegit a l'editor
  - 1b3. El sistema mostra les mètriques

### Diagrama de seqüència

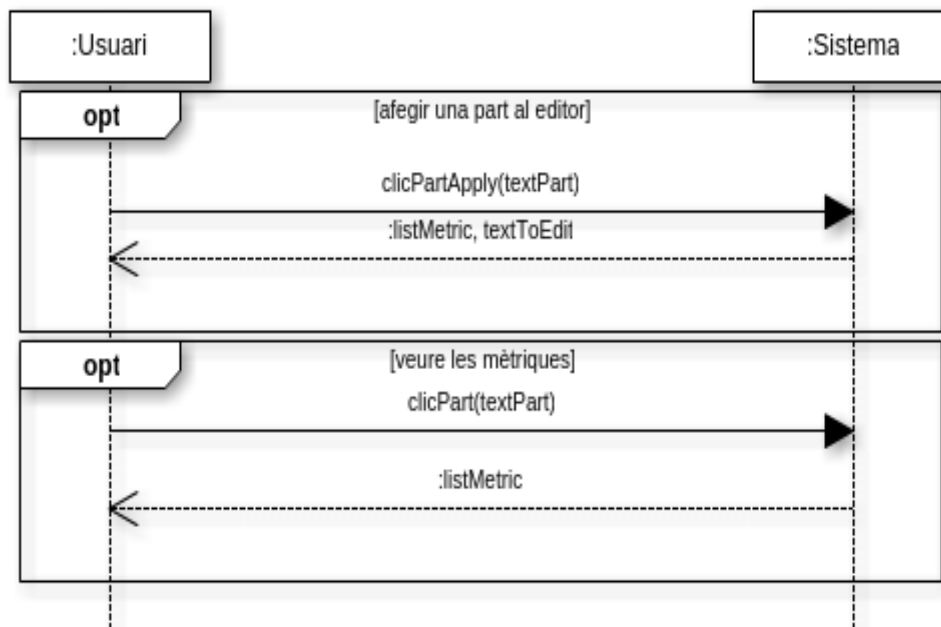


Diagrama de seqüència de cas d'ús Insertar una part a l'editor

### Contracte de l'operació.

#clicPartApply(textPart): listMetric, textToEdit

**Descripció:** L'esdeveniment mostra les mètriques que té una part d'un form d'un SRP i també afegeix la part a l'editor.

**Pre:** -

**post:** retorna les variables de la part (les mètriques) i els seus possibles valors. També torna la informació de l'editor canviada, ja que afegeix la part del SRP

#clicPart(textPart): listMetric

**Descripció:** L'esdeveniment mostra les mètriques que té una part d'un form d'un SRP

**pre:** -

**post:** retorna unes variables de la part (les mètriques) i els seus possibles valors

**Operació:** clicPartApply(textPart): listMetric, textToEdit

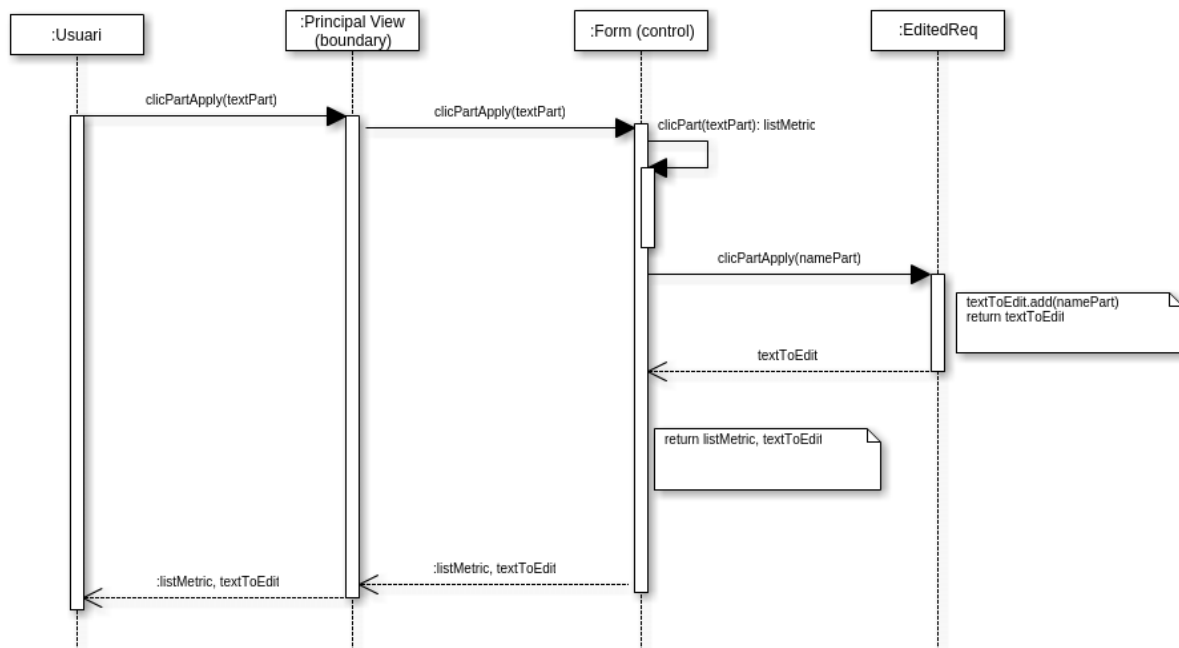


Diagrama de seqüència de l'operació clicPartApply

**Operació:** clicPart(textPart): listMetric

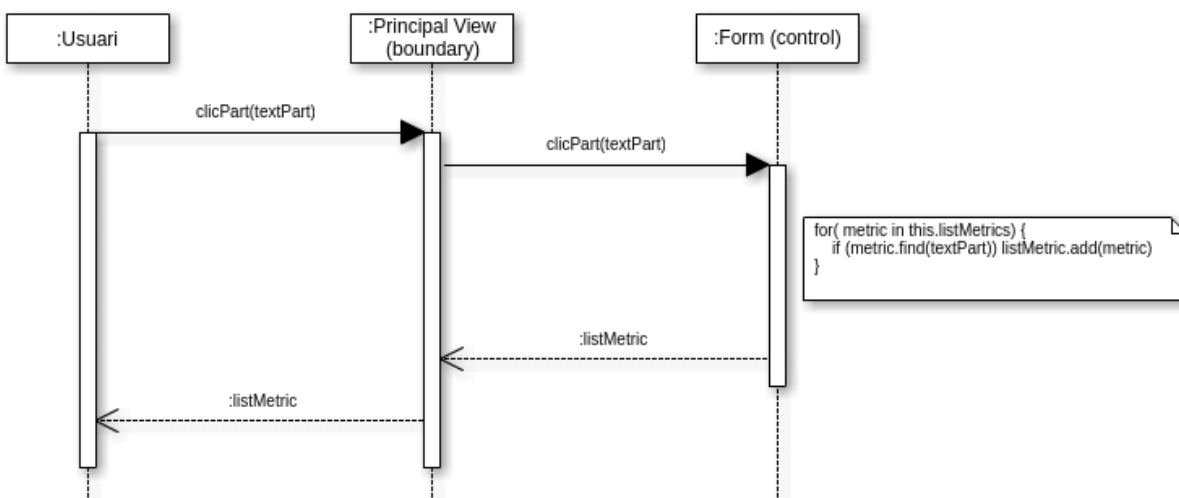


Diagrama de seqüència de l'operació clicPart



## 7 – Cas d'ús: Cercar els SRPs

### Descripció

L'usuari ha de poder fer cerques de paraules per poder trobar amb aquestes paraules clau, els patrons que contenen aquestes paraules.

### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol fer recerca de paraules.

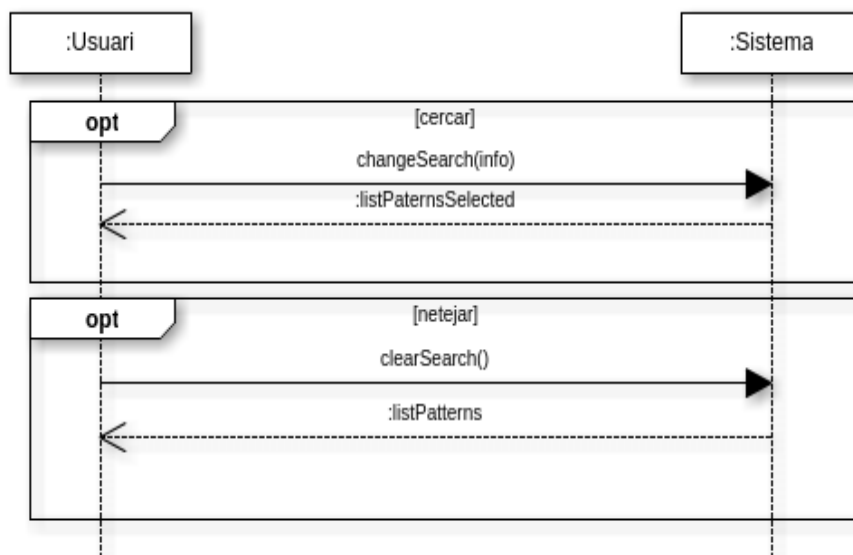
**Escenari d'èxit principal:**

1. L'usuari gestiona la cerca/ordenació dels SRPs.
2. S'acaba el cas d'ús.

**Extensions:**

- 1a. L'usuari vol cercar una paraula
  - 1a1. L'usuari escriu una paraula o paraules
  - 1a2. L'usuari cerca la paraula o paraules.
  - 1b3. El sistema mostra els SRPs que té aquest contingut.
- 1b. L'usuari vol netejar la recerca.
  - 1b1 L'usuari selecciona que vol netejar la recerca
  - 1b2 El sistema mostra tots els SRPs

### Diagrama de seqüència



### Contracte de l'operació.

#changeSearch(info)

**Descripció:** Aquest esdeveniment és per cercar paraules

**pre:** -

**post:** ListPatternsSelected, retorna els noms dels SRPs que contenen la informació que es vol buscar

#clearSearch()

**Descripció:** Aquest esdeveniment neteja la informació dels SRPs

**pre:**

**post:** listPatterns, retorna tots els SRPs

**Operació:** changeSearch(info)

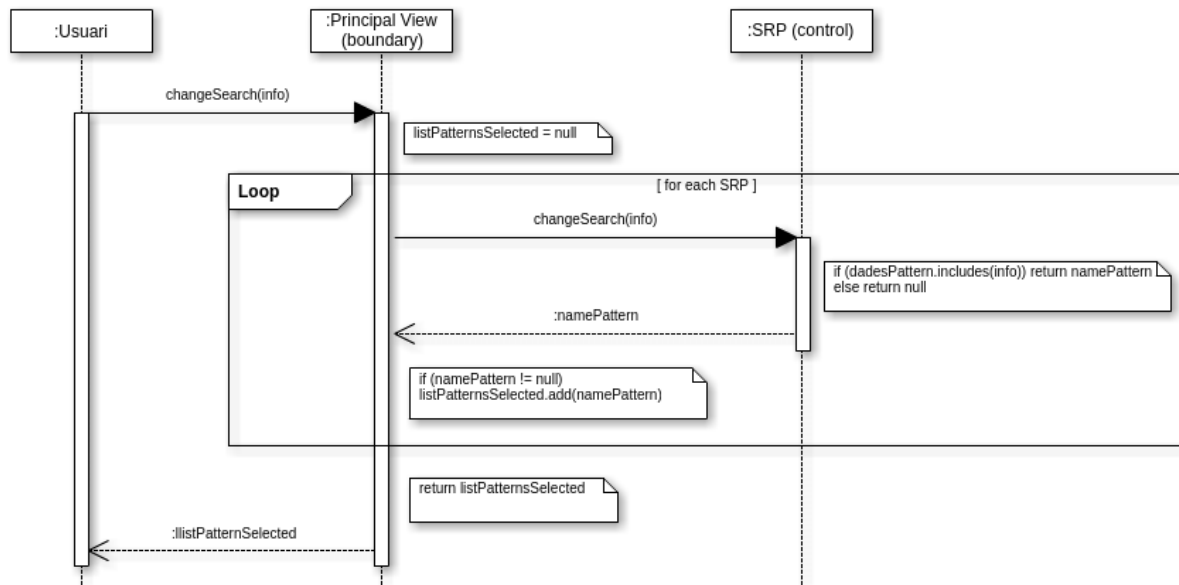


Diagrama de seqüència de l'operació changeSearch

**Operació:** clearSearch()

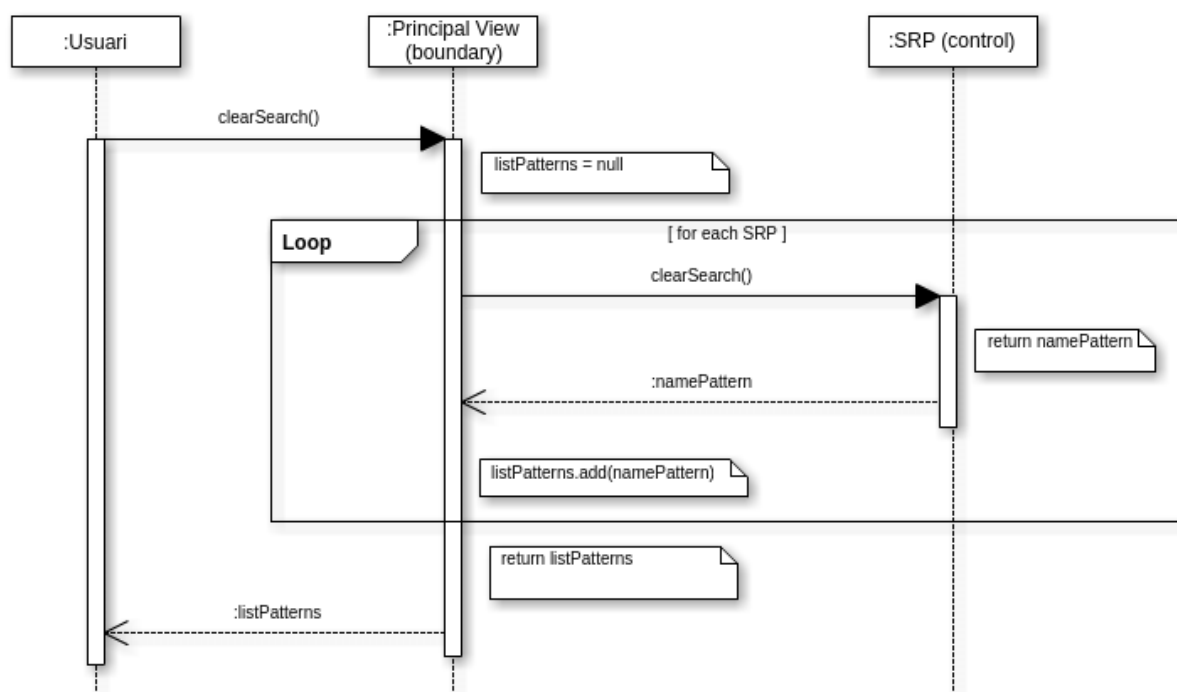


Diagrama de seqüència de l'operació clearSearch

## 8 – Cas d'ús: Seleccionar esquema

### Descripció

L'usuari ha de poder seleccionar una opció de l'esquema que es vol visualitzar.

### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol fer recerca de paraules.

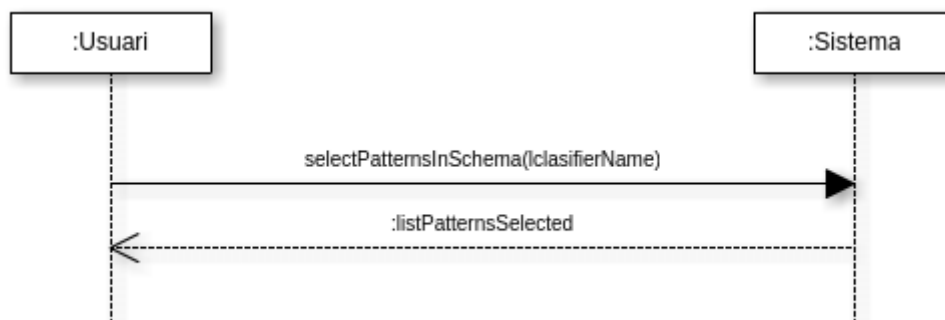
**Escenari d'èxit principal:**

1. L'usuari selecciona una opció de l'esquema.
2. El sistema mostra els SRPs d'aquella opció
3. S'acaba el cas d'ús.

**Extensions:**

-

### Diagrama de seqüència



*Diagrama de seqüència de cas d'ús Seleccionar esquema*

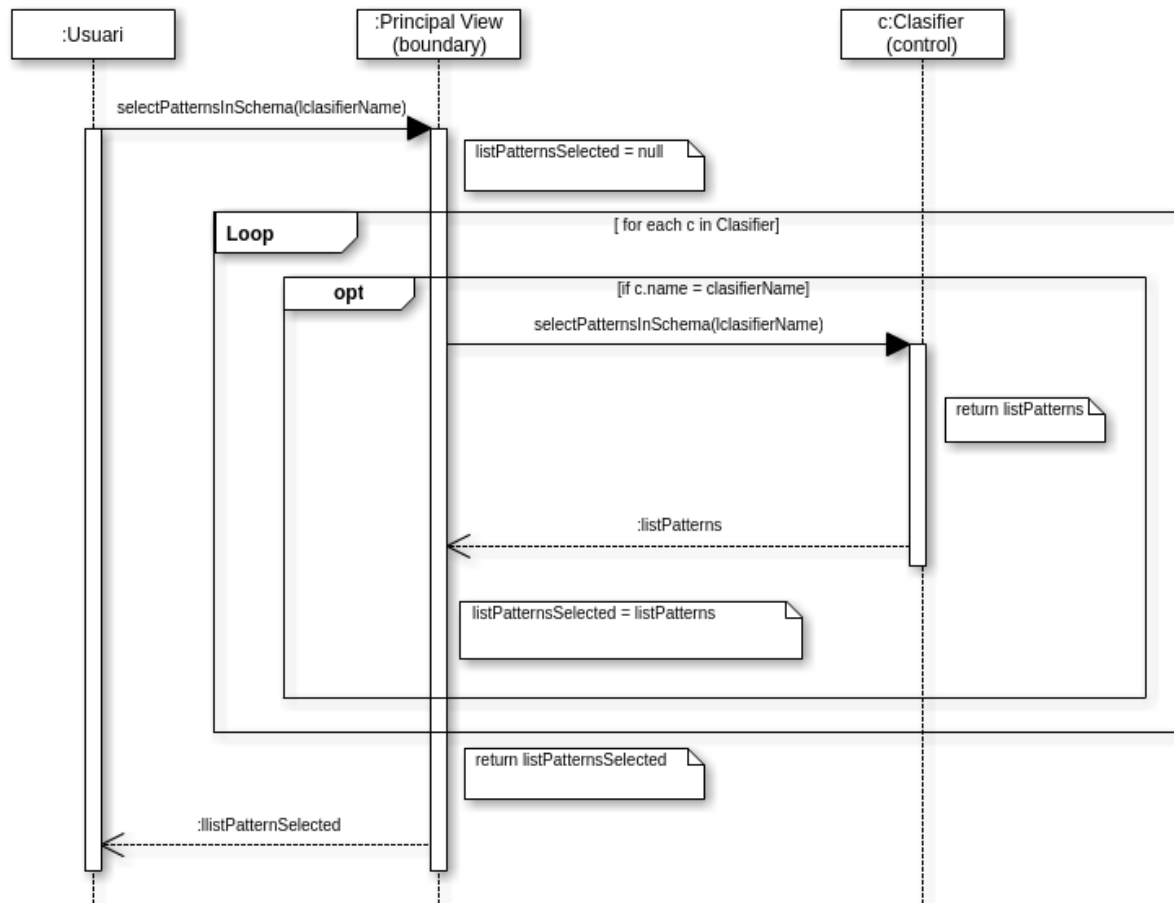
### Contracte de l'operació.

#selectPatternInSchema(clasifierName):listPatternsSelected

**Descripció:** Aquest esdeveniment selecciona un rootClasifier (un classificador del primer nivell) de l'esquema que usem i mostra els SRPs que conté aquest classificador

**pre:** -

**post:** torna els noms dels SRPs que conté el classificador seleccionat

**Operació: selectPatternInSchema(clasifierName):listPatternsSelected***Diagrama de seqüència de l'operació selectPatternInSchema*

## 9 – Cas d'ús: Obrir templates

### Descripció

L'usuari ha de poder obrir projectes prefabricats de l'aplicació

### Especificació

**Actor principal:** Usuari (enginyers de requisits, persones que treballen en empreses de desenvolupament de software)

**Pre-condició:** -

**Disparador:** L'usuari vol visualitzar projectes ja predefinit

**Escenari d'èxit principal:**

1. L'usuari selecciona el projecte que vol
2. S'acaba el cas d'ús.

**Extensions:**

- 1a. El sistema mostra una finestra amb un avís de confirmació.
  - 1a1. L'usuari ho confirma.
  - 1a2. El sistema mostra la informació en l'editor de text.

## Diagrama de seqüència

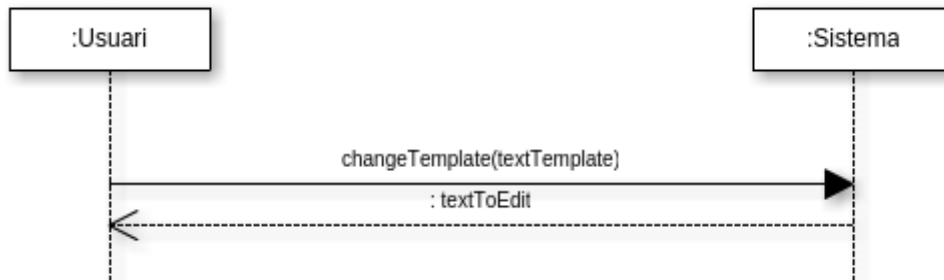


Diagrama de seqüència de cas d'ús Obrir templates

### Contracte de l'operació.

#changeTemplate(textTemplate): textToEdit

**Descripció:** L'esdeveniment selecciona un projecte ja prefabricat i mostra el seu contingut en l'editor de text.

**pre:** -

**post:** retorna el contingut que hi ha en un projecte seleccionat

### Operació: changeTemplate(textTemplate): textToEdit

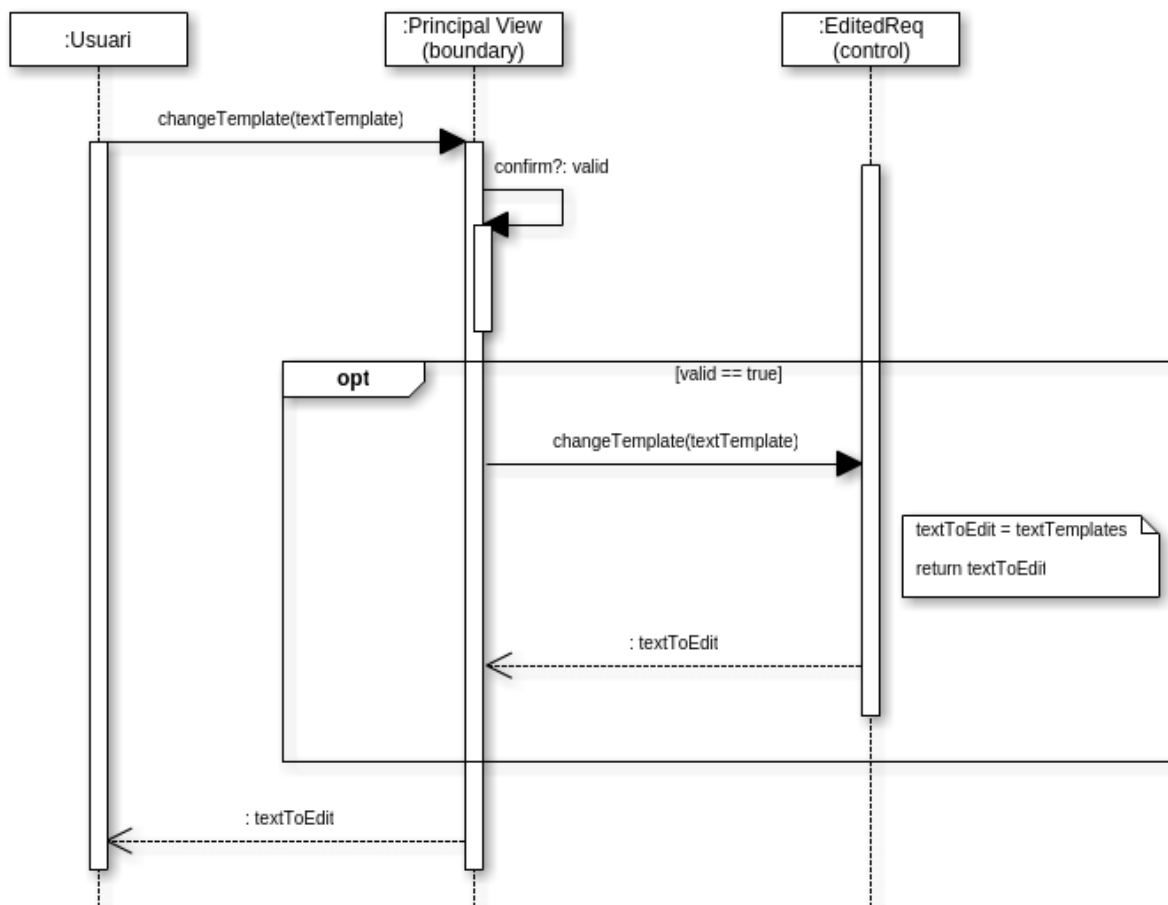


Diagrama de seqüència de l'operació changeTemplate

## Annex: Manual d'instal·lació

### **PABRE-RW**

L'aplicació PABRE-RW permet treballar els requisits d'un projecte de software, recuperant-los del servidor extern PABRE.

Diversos tipus de requisits estàndard s'emmagatzemen en el servidor extern (PABRE-WS), organitzats amb estructura SRP (Software Requirement Patterns: Patrons de Requisits de Software).

L'usuari pot:

- definir els seus propis projectes: romandran emmagatzemats en el seu propi ordinador.
- editar-los dins PABRE- RW
- afegir requisits recuperats del servidor extern.

El servidor PABRE es troba a la UPC (Universitat Politècnica de Catalunya) i és administrat pel grup GESSI.

### **Requeriments del sistema**

Tot el que cal per instal·lar PABRE- RW és:

- una *connexió a Internet*.
- un navegador *Google Chrome* instal·lat a l'ordinador.

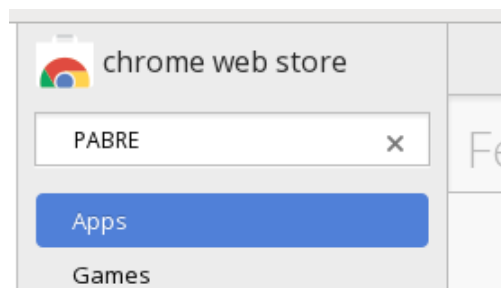
Qualsevol sistema operatiu en qualsevol equip farà el fet mentre es compleixin aquestes condicions.

**Precaució:** assegureu-vos que el vostre navegador és Google Chrome, no és possible instal·lar l'aplicació en cap altre navegador.

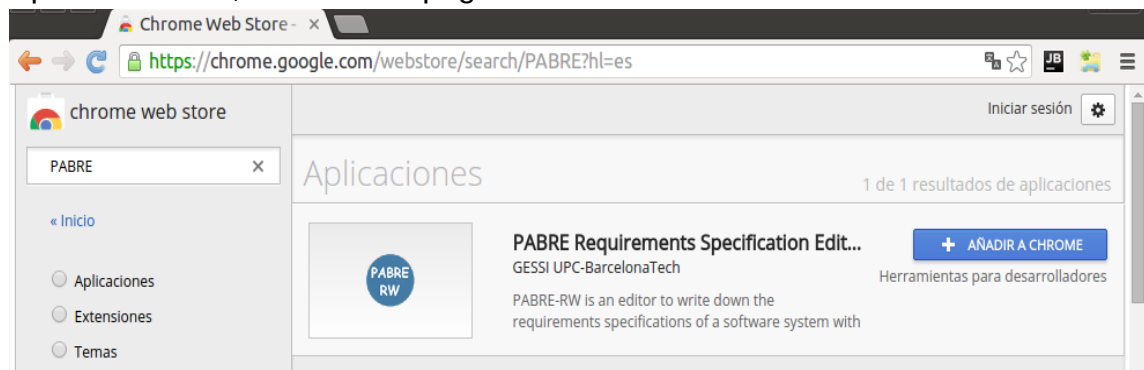
### **Instal·lació**

Per instal·lar l'app es pot obrir <https://chrome.google.com/webstore/category/apps> o bé, cercar **chrome webstore** en qualsevol cercador.

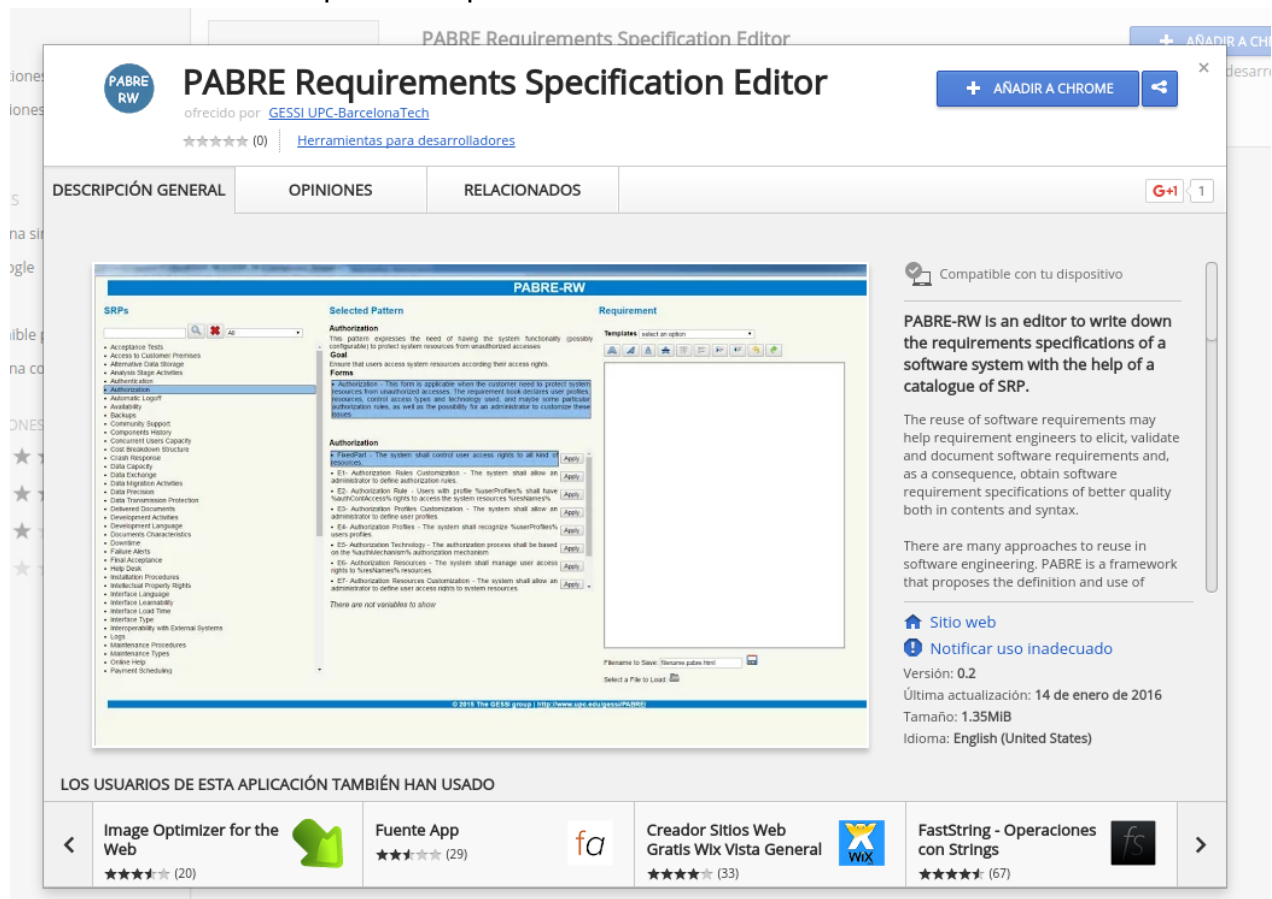
Un cop dins del *Chrome Webstore*, cal cercar PABRE-RW. Es pot fer escrivint la paraula **PABRE** al quadre de cerca que apareix a la cantonada superior esquerra de la web, si bé es pot introduir també **PABRE-RW** o el nom complet (**PABRE Requirements Specification Editor**).



Un cop feta la cerca, s'arriba a la pàgina:



On l'usuari pot instal·lar directament l'app prement el botó **+ AÑADIR A CHROME** o bé fer clic en una altra part de la zona gris clara, que correspòn a l'aplicació PABRE-RW, i així obtenir la descripció de l'aplicació:



en la qual també hi ha un botó **+ AÑADIR A CHROME** per instal·lar l'app directament. Per assegurar l'èxit de la instal·lació, convé comprovar que entre les apps del vostre

Chrome hi apareix la icona de PABRE-RW:



## Annex: User manual

### 1 Introduction

This app, PABRE-RW, allows you to work on software project requirements, retrieving them from a server.

All kind of standard requirements are stored in an external server (PABRE), organized under Software Requirement Patterns.

You can:

- define your own projects, that will remain stored in your computer
- edit them within PABRE- RW
- add them requirements retrieved from the server PABRE.

### 2 Starting Pabre-RW

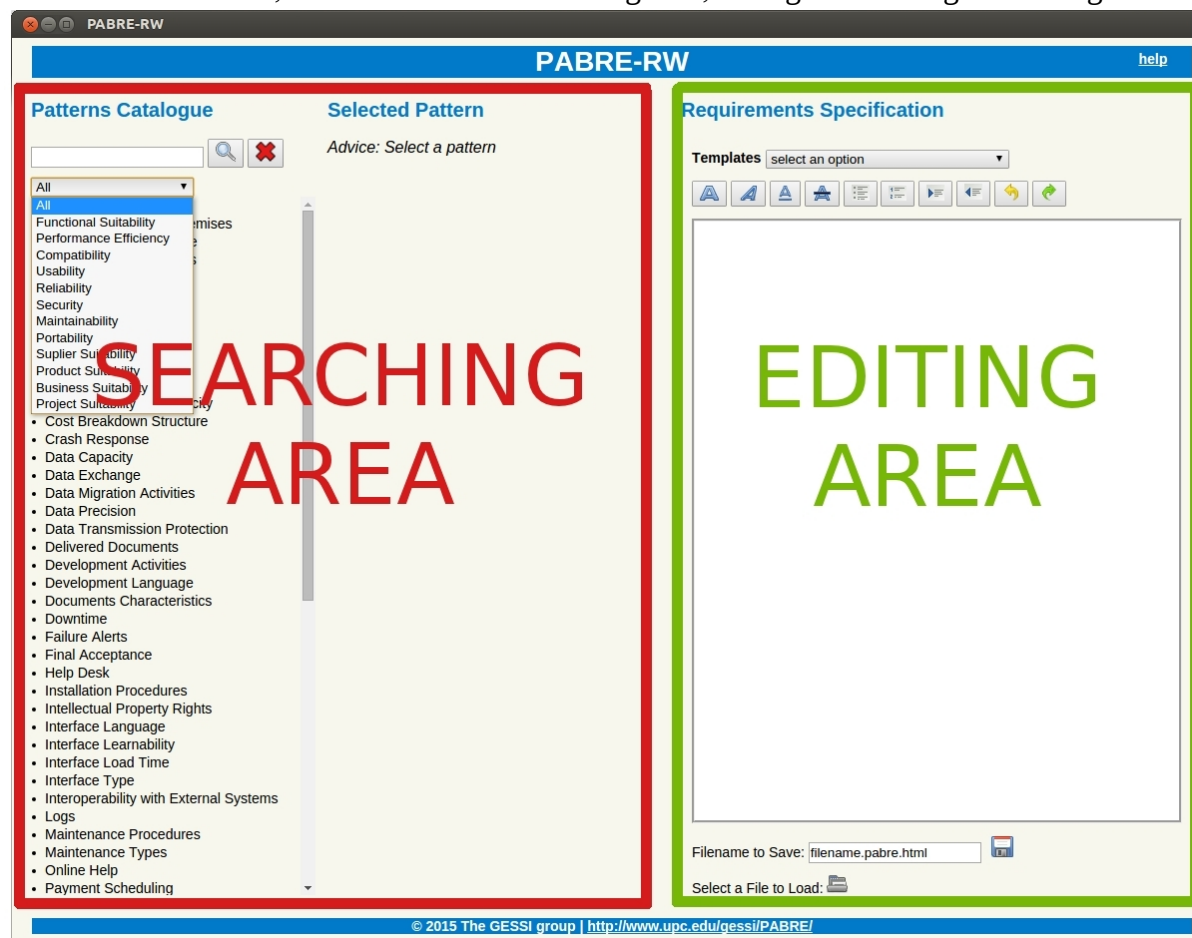
To start the app PABRE-RW open the Chrome's "my apps". The window of the app will open when you click the icon:



### 3 First steps

When PABRE-RW starts, a window opens with a blank project.

Two areas are shown, the left one is the searching area, the right one being the editing area:





You can build your project by filling the editor (big white zone in the editing area) with the requirements you need.

Three methods of inserting text are possible:

- Just *write text* on the editor.
- Insert *previously searched* requirements:  
see section **4** about searching and section **5.1** about inserting.
- Open a *template*: see section **5.5**

Opening templates is recommended for first-time users.

You can also edit, save and reload a project. See chapter **5**.

## 4 Searching through PABRE data

Two vertical areas exist in the search zone: **Patterns Catalogue** and **Selected Pattern**.

A first pattern search can be made on the left **Patterns Catalogue** zone.

When a pattern is clicked, one or more forms appear on the **Selected Pattern** zone.

Those forms can be expanded into its parts within the **Selected Pattern** zone.

### 4.1 Patterns Catalogue search

When just started, all patterns from PABRE sever are shown in the **Patterns Catalogue** zone.

You can click the one you wish: its forms will appear in the **Selected Pattern** zone.

Problem is, sequential search can be cumbersome. Two methods can facilitate the search:

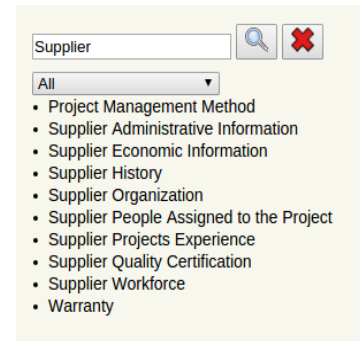
#### **Filter:**

In this method you type the text you wish to search for,

**Supplier** in the example, and click on 

The list of **Patterns** related to **Supplier** will be shown.

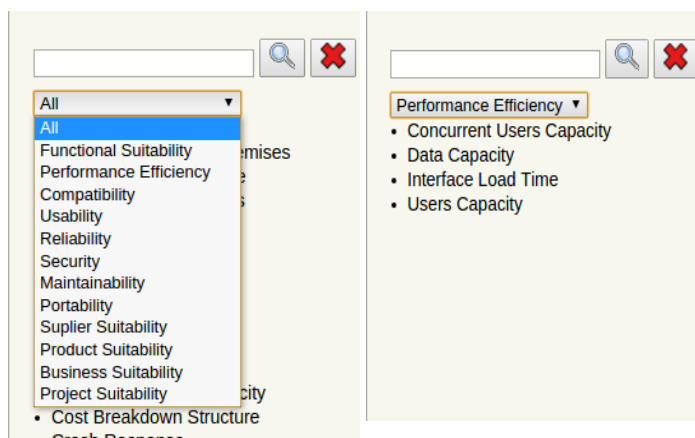
Note that the entered text not necessarily appears in the names of the **Patterns**. But all of them are related to **Supplier**, the word will appear in its forms or its parts.




#### **Classifier:**

In this method you open the list of classifiers (shown left) by clicking on ▼

Once the desired classifier is clicked, its related **Patterns** will be displayed (shown right)

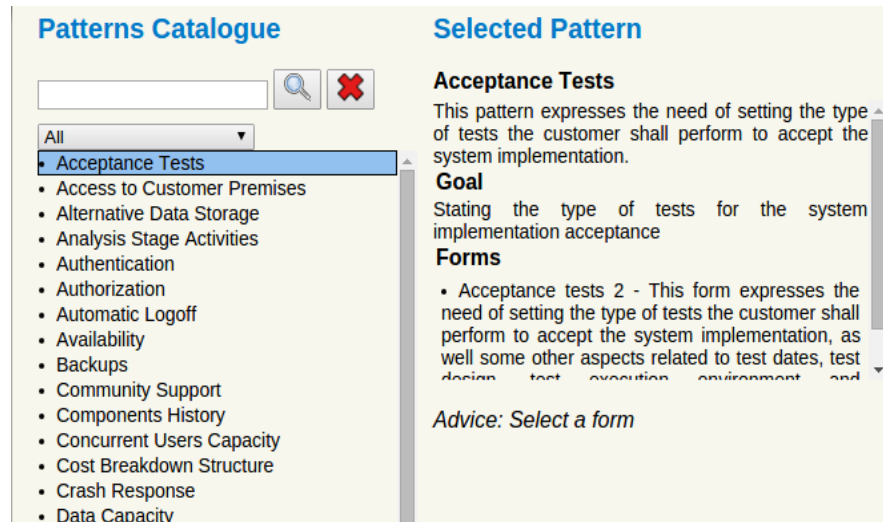


In any moment you can abandon the search performed and return to the initial list of all the patterns. That can be achieved either by selecting the **All** classifier or by clicking 

## 4.2 Forms search

Once a pattern is clicked (see section 4.1), its contents will be shown in the upper part of the **Selected Pattern** column:

In the example, the pattern **Acceptance Tests** has been clicked, as you can see because of its blue background. The **Selected Pattern** upper zone shows its goal and the form or forms included. You can scroll through them and click on the form you want to select. The background of the selected form becomes also blue.

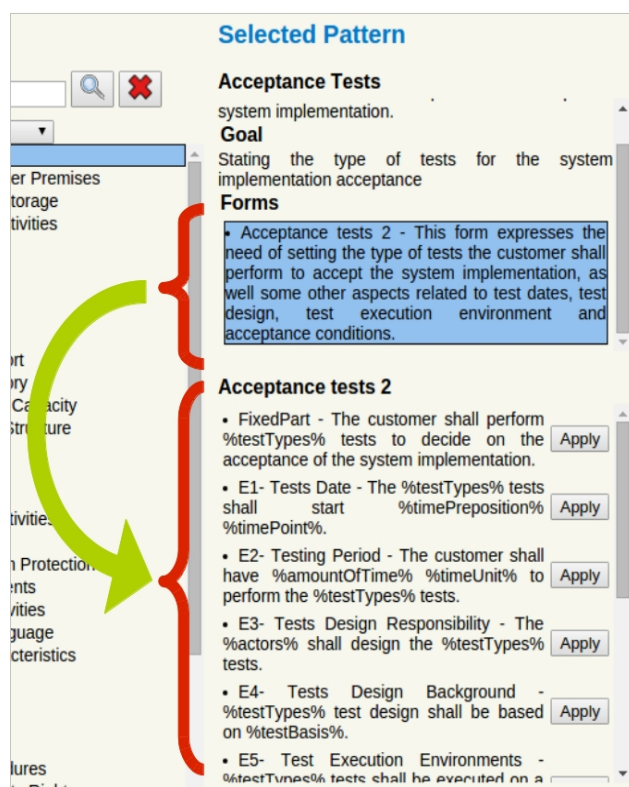


## 4.3 Parts search

Once a form is clicked (see section 4.2), its contents will be shown in the lower part of the **Selected Pattern** column, whereas the form background will become blue.

In the example, the **Acceptance test 2** form is selected and its parts are shown.

Every form contain one or more parts. A **Fixed Part** is mandatory, some extended parts (**E1**, **E2**, ...) can exist. Every part, either fixed or extended, behaves the same way.



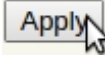
## 5 Edition

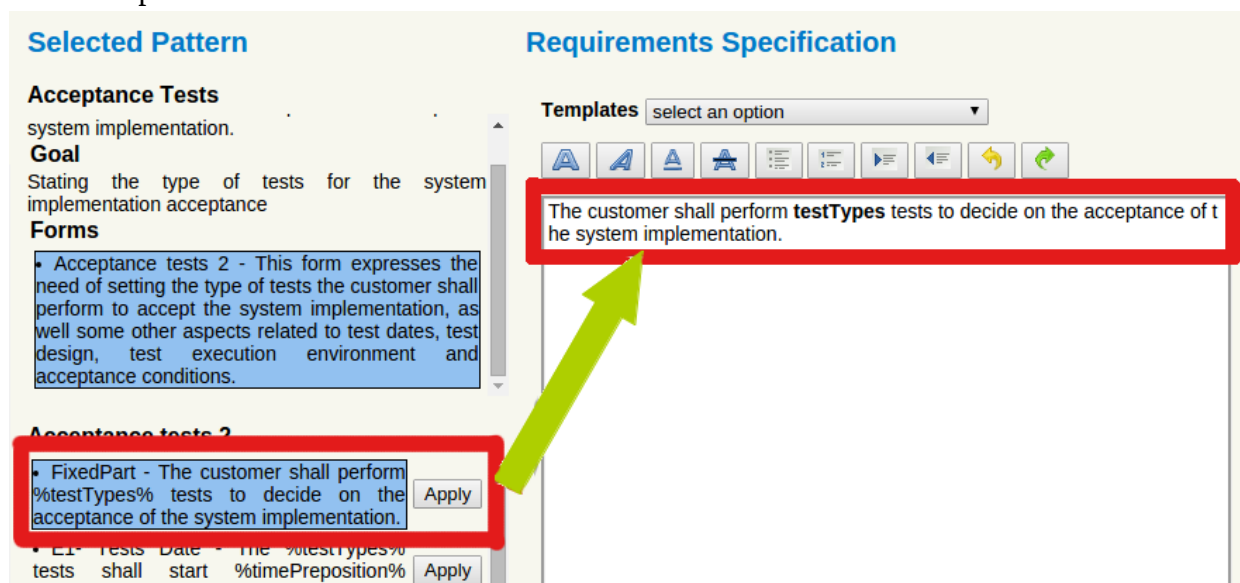
The big white area to the right is a basic text editor with some text formatting capacities.

That editor is where you will build your requirements and your project.

Its contents can be saved to a project file, and it is possible to open a previously saved file in this area.

### 5.1 Inserting requirements

When you click an  button on the central Selected Pattern area, its associated part will be copied to the editor:



Note that some expression inside the part appears in bold type in the editor. That expression matches the % enclosed text in the part: it must be adapted to necessities of the user. Some suggestions appear in a table on the bottom of the central Selected Pattern area, as explained above in section 4.3.

### 5.2 Editing buttons

The buttons just above the editing area form two groups: the eight to the left perform formatting actions on the text, whereas the two to the right are the usual do/undo:

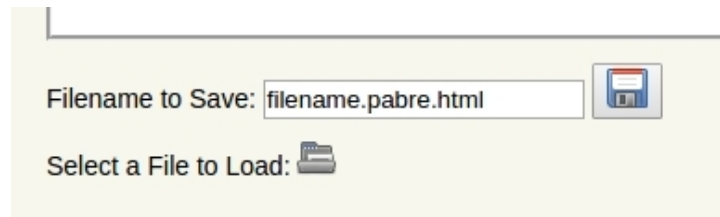



From left to right you can see: bold, italics, underline, strike through, bullets, numbers, indent, unindent, undo, redo.

All of them behave in the usual way.


### 5.3 Saving a project

To save the contents of the editor to a project file, you should look at the bottom of the editing area:



There, the system suggests you a name in **Filename to Save**. You can change that name. The extension `.pabre.html` is mandatory for the application to be able to retrieve it. Once the name is the right one you can save by pressing, the button 

### 5.4 Loading a saved project

A previously saved project file can be retrieved by clicking the  button at the very bottom of the editing area in **Select a File to Load**:


An auxiliary window will then open, in which you can explore through the file system and choose the file to open.

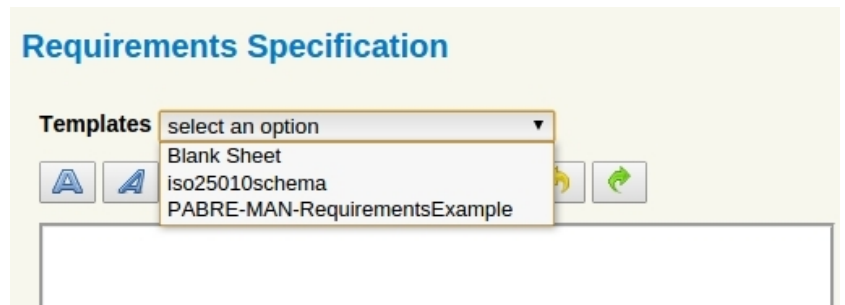
Be sure the chosen file has the extension `.pabre.html` or else an error will arise.

### 5.5 Opening a template

Some texts ready to insert into the editor, called **Templates**, are included in the app.

Some of them are special ready-made projects. They can be used as examples, as playground for training, or as the starting point of your own projects.

To open a template you must open the drop-down at the **Templates**, just above the editing buttons, by clicking on . When you click a template, its content will be displayed inside the editor.



There is a **Blank Sheet** template useful to quickly discard your project and begin a new one.